

Escuela Politécnica Superior

19
20

Trabajo fin de grado

Desarrollo de un plug-in para navegador web
para fomentar la transparencia periodística



Germán Darío Esquinazi Bachoer

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Desarrollo de un plug-in para navegador web para
fomentar la transparencia periodística**

**Autor: Germán Darío Esquinazi Bachoer
Tutor: Francisco Jurado Monroy**

abril 2020

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, n.º 1

Madrid, 28049

Spain

Germán Darío Esquinazi Bachoer

Desarrollo de un plug-in para navegador web para fomentar la transparencia periodística

Germán Darío Esquinazi Bachoer

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

AGRADECIMIENTOS

Ante nada, me gustaría agradecer y reconocer el trabajo de algunas personas sin las cuales no hubiese podido sacar este proyecto adelante. En primer lugar a mis padres, pues me han acompañado durante todo el proyecto, han servido como conejillos de indias en varias ocasiones y me han dado numerosos consejos para mejorar. En segundo lugar a mi novia Alicia, la cual ha aguantado todas mis quejas y momentos de desesperación y los ha contrarrestado con palabras de animo y apoyo.

Por último, pero no por ello menos importante, quiero agradecer a mi tutor Francisco todo el trabajo que ha realizado durante estos meses, respondiendo a mis numerosos correos de dudas y por no dejar de apoyarme hasta el final, a pesar de la situación vivida y las complicadas circunstancias.

RESUMEN

Cada día es más complejo discernir entre información falsa y verdadera. Cada día, las noticias se vuelven más complicadas de entender pues no solo hay que tener en cuenta el suceso ocurrido, sino en qué medio se está leyendo, quién lo está redactando, en qué contexto de noticias se está publicando, etc. Para poder comprender realmente una noticia, y obtener información objetiva y verídica, es necesario que el usuario realice un ejercicio de análisis de la noticia. Por ello, existe un movimiento en el mundo periodístico llamado Transparencia Periodística que pretende dotar al usuario de herramientas para realizar este análisis.

En este documento se detalla el desarrollo de un Trabajo Final de Grado en el que se pretende colaborar con este movimiento desarrollando una extensión para el navegador Chrome llamada *Transparency Plugin*. Esta herramienta provee al usuario de métodos para analizar en profundidad la noticia y obtener información relevante como son los autores, la fecha de publicación, los lugares mencionados, la carga emocional y subjetiva, o las entidades presentes en el texto.

El proyecto está desarrollado en Angular 8 para el lado del cliente y Flask para el lado del servidor. Además, se hacen uso de muchas herramientas de análisis de texto y procesamiento de lenguaje natural como son Newspaper3k, SpaCy y Textblob entre otros.

PALABRAS CLAVE

Aplicación web, Transparencia Periodística, Angular, Flask, Chrome, Extension

ABSTRACT

Every day it gets more complex to find whether certain information is true or false. Every day, the news gets harder to understand because we do not only need to have in mind the event itself, but also the media where is being read from, the author, the context of events that are happening at the same time, and so on. To comprehend a new and get the most out of it, the user needs to make an exercise of analysis. For that matter, there is a journalism movement called News Transparency which pretends to provide tools for users to make that analysis.

This document details the development of a Final Degree Project which aims to collaborate with this movement developing a Chrome extension called Transparency Plugin. This tool provides methods for the user to analyse news in-depth and extract relevant information such as authors, publication date, mentioned places, emotional burden and subjectivity levels of the text or entities present in the text.

The project is developed in Angular 8 for the client-side and Flask for the server-side. Also, it uses a lot of text analysis and natural language processing tools like Newspaper3k, SpaCy or Textblob.

KEYWORDS

Web application, News Transparency, Angular, Flask, Chrome, Plugin

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Organización de la memoria	2
2	Estado del Arte	3
2.1	TJ Tool	3
2.2	Azure API Text Analytics	5
2.3	Social Media Macroscope	5
2.4	Elección de Chrome como navegador	6
3	Análisis	7
3.1	Casos de Uso	7
3.2	Requisitos	9
3.2.1	Requisitos Funcionales	9
3.2.2	Requisitos No Funcionales	10
3.3	Elección de tecnologías	10
3.3.1	Tecnologías Frontend	10
3.3.2	Tecnologías Backend	12
3.4	Plataforma y herramientas de trabajo	14
3.4.1	Plataforma de trabajo	14
3.4.2	Herramientas de trabajo	14
3.5	Metodología de trabajo	15
4	Diseño y Desarrollo	17
4.1	Arquitectura de la aplicación	17
4.1.1	Frontend	17
4.1.2	Backend	19
4.1.3	Comunicación entre cliente y servidor	20
4.2	Maquetas	21
4.3	Desarrollo del proyecto	21
4.3.1	Detalles de implementación del cliente	21
4.4	Accesibilidad	24
5	Pruebas y Resultados	25

5.1 Pruebas Unitarias	25
5.2 Pruebas con usuarios	29
6 Conclusiones y Trabajo futuro	35
6.1 Conclusiones	35
6.2 Trabajo Futuro	36
Bibliografía	39
Apéndices	41
A Descripción de casos de uso	43
B Product Backlog	47
B.1 Iteración 1	47
B.2 Iteración 2	52
B.3 Iteración 3	57
C Arquitectura de la aplicación	63
D Maquetas	65
E Aplicación Final	69
F Informe de consentimiento de datos	81
G Cuestionario de Usabilidad	83
G.1 Preguntas destinadas al cálculo del SUS	83
G.2 Preguntas destinadas a medir la satisfacción	85
H Resultados del cuestionario de usabilidad	87
H.1 Respuestas para el SUS	87
H.2 Respuestas para la satisfacción	92

LISTAS

Lista de figuras

2.1	Plugin de Transparencia del Público	4
2.2	Azure API Text Analytics	5
3.1	Diagrama de Casos de Uso	8
4.1	Diagrama de secuencia	21
5.1	Resumen de Test unitario Fallido	28
5.2	Resumen de Test unitario Correcto	28
B.1	Product Backlog 1-10-19	47
B.2	Product Backlog 5-10-19	48
B.3	Product Backlog 6-10-19	48
B.4	Product Backlog 10-10-19	49
B.5	Product Backlog 2-11-19	49
B.6	Product Backlog 26-12-19	49
B.7	Product Backlog 30-12-19	50
B.8	Product Backlog 5-1-20	50
B.9	Product Backlog 7-1-20	50
B.10	Product Backlog 14-1-20	51
B.11	Product Backlog 24-1-20	51
B.12	Product Backlog 25-1-20	52
B.13	Product Backlog 26-1-20	53
B.14	Product Backlog 29-1-20	53
B.15	Product Backlog 6-2-20	54
B.16	Product Backlog 12-2-20	54
B.17	Product Backlog 13-2-20	55
B.18	Product Backlog 14-2-20	55
B.19	Product Backlog 16-2-20	56
B.20	Product Backlog 26-2-20	56
B.21	Product Backlog 30-2-20	57
B.22	Product Backlog 27-2-20	57
B.23	Product Backlog 28-2-20	58

B.24	Product Backlog 1-3-20	58
B.25	Product Backlog 2-3-20	59
B.26	Product Backlog 4-3-20	59
B.27	Product Backlog 7-3-20	60
B.28	Product Backlog 8-3-20	60
B.29	Product Backlog 10-3-20	61
B.30	Product Backlog 12-3-20	61
C.1	Diagrama de componentes de la aplicación	64
D.1	Maqueta - Página de inicio	65
D.2	Maqueta - Página de categorías	66
D.3	Maqueta - Categoría Datos Básicos	66
D.4	Maqueta - Categoría Análisis Emocional	67
D.5	Maqueta - Categoría Análisis Geográfico	67
D.6	Maqueta - Categoría Noticia Accesible	68
D.7	Maqueta - Categoría Análisis Texto	68
E.1	Producto Final - Plugin	69
E.2	Producto Final - Página de inicio	70
E.3	Producto Final - Página de selección de categorías	71
E.4	Producto Final - Categoría de Datos Básicos 1	72
E.5	Producto Final - Categoría de Datos Básicos 2	73
E.6	Producto Final - Categoría de Datos Básicos 3	74
E.7	Producto Final - Categoría de Análisis Geográfico	75
E.8	Producto Final - Categoría de Análisis Emocional	76
E.9	Producto Final - Modo Alto Contraste 1	77
E.10	Producto Final - Modo Alto Contraste 2	78
E.11	Producto Final - Categoría de Noticia Accesible	79
E.12	Producto Final - Categoría de Análisis del Texto	80
G.1	Parte general del cuestionario	84
G.2	Parte específica 1 del cuestionario	85
G.3	Parte específica 2 del cuestionario	86
H.1	Resultado pregunta 1 del cuestionario	87
H.2	Resultado pregunta 2 del cuestionario	87
H.3	Resultado pregunta 3 del cuestionario	88
H.4	Resultado pregunta 4 del cuestionario	88
H.5	Resultado pregunta 5 del cuestionario	89

H.6	Resultado pregunta 6 del cuestionario	89
H.7	Resultado pregunta 7 del cuestionario	90
H.8	Resultado pregunta 8 del cuestionario	90
H.9	Resultado pregunta 9 del cuestionario	91
H.10	Resultado pregunta 10 del cuestionario	91
H.11	Resultados pregunta 11 del cuestionario	92
H.12	Resultados pregunta 11 del cuestionario	93
H.13	Resultados pregunta 11 del cuestionario	94
H.14	Resultados pregunta 11 del cuestionario	95
H.15	Resultado pregunta 12 del cuestionario	95
H.16	Resultado pregunta 13 del cuestionario	96
H.17	Resultado pregunta 14 del cuestionario	96
H.18	Resultado pregunta 15 del cuestionario	97
H.19	Resultado pregunta 16 del cuestionario	97
H.20	Resultado pregunta 17 del cuestionario	98
H.21	Resultado pregunta 18 del cuestionario	98
H.22	Resultado pregunta 19 del cuestionario	99
H.23	Resultado pregunta 20 del cuestionario	99
H.24	Resultado pregunta 21 del cuestionario	100
H.25	Resultado pregunta 22 del cuestionario	100
H.26	Resultado pregunta 23 del cuestionario	101
H.27	Resultado pregunta 24 del cuestionario	101
H.28	Resultado pregunta 25 del cuestionario	102
H.29	Resultado pregunta 26 del cuestionario	102
H.30	Resultado pregunta 27 del cuestionario	103

Lista de tablas

2.1	Estadísticas de uso de navegadores web	6
3.1	Planificación del proyecto	16
5.1	Resultado de parte específica de cuestionario de usabilidad	32
A.1	Caso de Uso - 01	43
A.2	Caso de Uso - 02	44
A.3	Caso de Uso - 03	44
A.4	Caso de Uso - 04	45
A.5	Caso de Uso - 05	45

INTRODUCCIÓN

1.1. Motivación

La motivación para el desarrollo de Transparency Plugin vino por dos fuentes diferentes. Día tras día, los medios de comunicación se vuelven más complicados, cada vez hay más información que se debe inferir de la noticia por el contexto en el que se desarrolla. De esta manera, existe mucha información en una noticia que no se ve a simple vista y que requiere de un análisis más profundo para ser entendida.

Es así como surge el término de transparencia periodística. La transparencia periodística permite al lector evaluar la calidad de la información presentada, le provee de herramientas que ayudan a contrastar dicha información y le permite entenderla mejor. Por ello, la motivación principal de este proyecto era construir una aplicación que ayudase a analizar esta transparencia periodística en todas las noticias. [1]

La otra fuente de inspiración, a modo de motivación personal, era mi búsqueda de un proyecto que me obligase a aprender algo nuevo relacionado con el desarrollo web, pues es a donde quiero orientar mi carrera profesional. Además, quería desarrollar algo que estuviese fuertemente relacionado con la usabilidad y accesibilidad. Es por ello que podemos ver en el proyecto una especial atención a estos dos aspectos.

1.2. Objetivos

El objetivo principal de la aplicación desarrollada es dotar al usuario de herramientas que le permitan analizar cualquier noticia y evaluar el grado de transparencia periodística. Para ello, la aplicación provee al usuario de distintas herramientas con las que podrá extraer más información del texto. Para poder alcanzar el objetivo principal, se han abordado los siguientes objetivos específicos:

Datos Básicos: Este objetivo permitirá obtener toda la información básica de la noticia como son autores, fecha de publicación, resumen, palabras clave, imagen de portada, lista de

vídeos presentes en la noticia y una lista de entidades presentes en el texto.

Análisis Emocional: Este objetivo permitirá realiza un análisis del lenguaje de la noticia para extraer información acerca de las emociones presentes. De esta manera, podemos saber si esta hablando a favor o en contra, el nivel de subjetividad aportado por el autor y el nivel de carga emocional dividido en 12 emociones principales.

Análisis Geográfico: Este objetivo permitirá extraer la información de los lugares mencionados en la noticia y representarlos en un mapa interactivo.

Análisis del texto: Este objetivo permitirá analizar el texto en busca de entidades presentes y remarcarlas en el texto clasificándolas en uno de estos tipos: personas, localizaciones, organizaciones y misceláneo.

Usabilidad y Accesibilidad: Este objetivo permitirá que usuarios que no puedan navegar por la pagina web de la noticia por una mala construcción de su código, puedan navegar por una página simplificada con toda la información importante de la noticia, asegurando la correcta construcción y estructuración del código para que cualquier lector web pueda navegar fácilmente por ahí.

1.3. Organización de la memoria

Este documento esta compuesto de 6 capítulos en los cuales se detallan las fases de desarrollo seguidas durante el proyecto.

En el capítulo 2 se detallarán el estado del arte en el que se presentarán ejemplos de aplicaciones con funciones similares a la desarrollada.

En el capítulo 3 se expondrá la fase de análisis del proyecto, donde se definirán cosas como los casos de uso, requisitos, metodología de trabajo y las tecnologías usadas.

En el capítulo 4 se expondrá las fases de diseño y desarrollo del proyecto, donde se definirán la arquitectura de la aplicación, las maquetas y ciertas decisiones de diseño tomadas durante el desarrollo que cabe mencionar. Además, existe una sección dedicada al apartado de accesibilidad en la aplicación.

En el capítulo 5 se detallarán las pruebas realizadas durante la fase de pruebas. Esto incluye pruebas unitarias y pruebas con usuarios. También se expondrán los resultados de estas pruebas.

En el capítulo 6 se expondrán las conclusiones finales de la memoria y el proyecto así como ideas para un futuro desarrollo de la aplicación.

ESTADO DEL ARTE

En esta sección detallaremos aplicaciones y herramientas similares o que realizan parte de la función que desempeña Tansparency Plugin. Por último, existe un apartado acerca de la elección del navegador web con el que se puede usar el plugin.

2.1. TJ Tool

La primera herramienta que debemos ver es el plugin que posee la web del periódico Público [2]. TJ Tool pretende aportar transparencia periodística a la noticia otorgándole trazabilidad a la información. Esta herramienta otorga un porcentaje de transparencia a la noticia según los siguientes parámetros:

- Hacer pública la política editorial.
- Explicitar el lugar de redacción.
- Citar las fuentes.
- Aportar documentos o enlaces de contexto.
- Explicar el motivo de redacción de la noticia.
- Mostrar el/la autor/a de la noticia.
- Adjuntar documentación de apoyo.
- Indicar la fecha de redacción y modificación de la noticia.

En la figura 2.1 podemos ver la herramienta analizando una noticia.

El coronavirus provoca la mayor subida del paro de la historia: 302.365 desempleados más

Esta noticia tiene un valor de transparencia de 82 sobre un máximo de 100.

Este es el resultado generado por la Calculadora de Transparencia de nuestra herramienta. [Aquí](#) puedes encontrar más información sobre los criterios de transparencia aplicados por la Herramienta de Periodismo Transparente de Público.

Esta noticia ha sido actualizada desde su creación. Para ver el Mapa de Transparencia de las actualizaciones anteriores, selecciona la versión que quieres consultar aquí:

Actualización de la noticia: 02/04/2020 10:38 (Versión: 3012751)

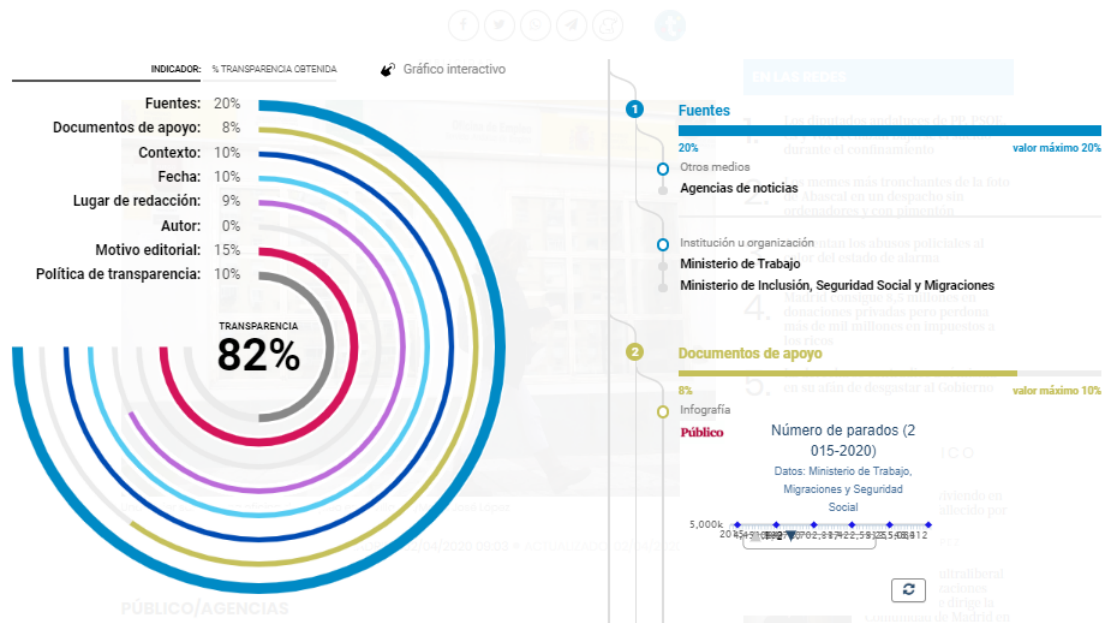


Figura 2.1: Plugin de Transparencia del periódico Público

2.2. Azure API Text Analytics

La API de análisis de texto de Azure [3] es una herramienta de pago de Microsoft la cual realiza un análisis de un texto y lo devuelve en formato JSON . Aunque no proporciona un análisis de la transparencia periodística, este API proporciona lo siguiente:

- Detección de idioma.
- Listado de palabras clave.
- Detección y clasificación de entidades.
- Análisis de la opinión general del texto.
- Análisis de la opinión de cada frase del texto.

En la figura 2.2 podemos ver un ejemplo de la herramienta. Este ejemplo está sacado de su página web Microsoft Azure.



Figura 2.2: Herramienta de análisis de texto de Azure

2.3. Social Media Macroscope

Social Media Macroscope [4] es un proyecto desarrollado por la Universidad de Illinois y el National Center for Supercomputing Applications (NCSA) cuyo objetivo es el análisis de datos de redes sociales y su visualización de manera accesible para investigadores y alumnos.

Aunque tampoco ha sido pensada para analizar la transparencia periodística, la herramienta Social Media Macroscope puede emplearse para realizar un análisis sobre los datos y obtener conclusiones como phrase mining, análisis de sentimientos, análisis de la red de contactos y clasificación por machine-learning.

2.4. Elección de Chrome como navegador

Debido a las limitaciones de la herramienta, el soporte de la aplicación para varios navegadores tiene que ser desarrollado por separado para cada uno. Por ello, se decidió centrarse en un solo navegador al principio, dejando los demás navegadores para futuros desarrollos.

La elección de que navegador escoger estuvo basada en 2 factores principalmente. El primero fue el uso mayoritario de la población. Como podemos ver en la tabla 2.1, Chrome es el navegador más usado por una amplia diferencia. El segundo factor fue la facilidad de desarrollo de extensiones. En este aspecto, Chrome resultó ser el más sencillo de implementar e integrar con las tecnologías escogidas para el desarrollo de la aplicación. Además, cuenta con una amplia documentación para desarrollo de extensiones con Angular [5], cosa de la que carecen otros navegadores [6].

Navegador	StatCounter Octubre 2019	NetMarketShare Noviembre 2019	Wikimedia Noviembre 2019
Chrome	64.92 %	63.82 %	48.7 %
Safari	15.97 %	18.68 %	22.0 %
Firefox	4.33 %	3.71 %	4.9 %
Samsung Internet	3.29 %	2.39 %	2.7 %
UC	2.94 %	0.82 %	0.3 %
Opera	2.34 %	0.79 %	1.1 %
Edge	2.05 %	2.56 %	1.9 %
IE	1.98 %	2.90 %	3.9 %
AOSP	0.59 %	0.52 %	0.2 %
Otros	1.59 %	3.81 %	14.3 %

Tabla 2.1: Estadísticas de uso de navegadores web obtenidas de StatCounter, NetMarketShare y Wikimedia [7]

ANÁLISIS

En este capítulo se detallarán todos los aspectos referentes a la fase de análisis del proyecto. Se definirán los casos de uso, los requisitos, las tecnologías usadas y la metodología de trabajo seguida.

3.1. Casos de Uso

La figura 3.1 representa el diagrama de casos de uso de la aplicación. En él se muestran las principales acciones que puede realizar el usuario. Como podemos observar en el diagrama, es necesario haber analizado una Url previamente para poder acceder a las distintas categorías de la aplicación. En el Anexo A se encuentran listados los principales casos de uso de la aplicación. Estos casos de uso son genéricos y son válidos en diversas situaciones y representan las principales acciones que puede hacer un usuario para interactuar con la aplicación.

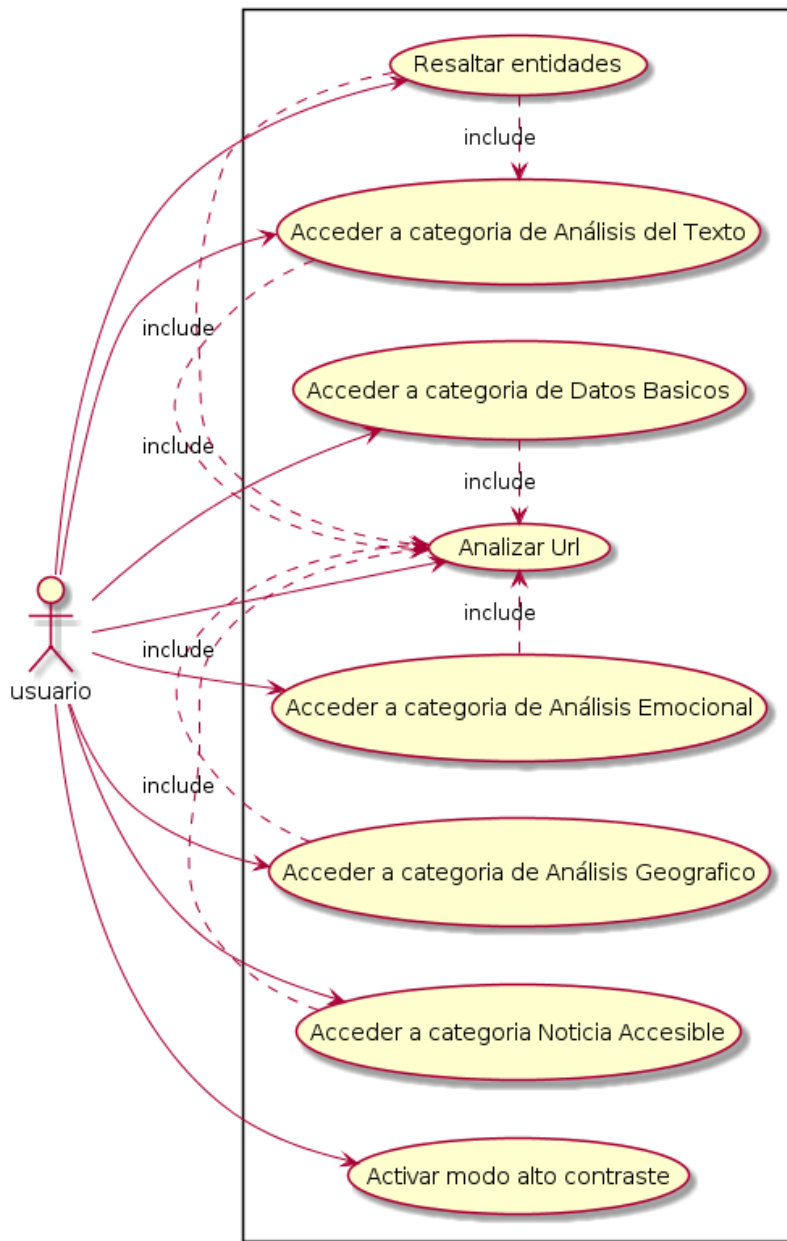


Figura 3.1: Diagrama de casos de uso de la aplicación

3.2. Requisitos

En esta sección definiremos los requisitos funcionales y no funcionales de la aplicación.

3.2.1. Requisitos Funcionales

RF-1.— El usuario deberá ser capaz de indicarle a la aplicación qué noticia desea analizar a través de una URL.

RF-1.1.— El usuario podrá introducir la URL en un formulario o analizar la página en la que se encuentre actualmente.

RF-2.— La aplicación deberá organizar la información que desea mostrar al usuario en categorías.

RF-3.— El usuario deberá poder elegir que categoría desea ver.

RF-4.— La aplicación tendrá las siguientes categorías:

RF-4.1.— Datos básicos.

RF-4.2.— Análisis emocional.

RF-4.3.— Análisis geográfico.

RF-4.4.— Noticia Accesible.

RF-4.5.— Análisis del texto.

RF-5.— La categoría de Datos Básicos contendrá la siguiente información:

RF-5.1.— Autores de la noticia.

RF-5.2.— Fecha de publicación de la noticia.

RF-5.3.— Palabras clave de la noticia.

RF-5.4.— Resumen de la noticia.

RF-5.5.— La imagen principal de la noticia, si existe.

RF-5.6.— Un listado de vídeos de la noticia, si existen.

RF-5.7.— Un listado de entidades presentes en el texto.

RF-6.— La categoría de Análisis emocional contendrá la siguiente información:

RF-6.1.— Grado de la polaridad de la noticia.

RF-6.2.— Grado de la subjetividad de la noticia.

RF-6.3.— Grado de sentimiento de la noticia. Dichos sentimiento es medido a través de las siguientes categorías:

RF-6.3.1.— Ira.

RF-6.3.2.— Anticipación.

RF-6.3.3.— Disgusto.

RF-6.3.4.— Miedo.

RF-6.3.5.— Disfrute.

RF-6.3.6.— Negatividad.

RF-6.3.7.— Positividad.

RF-6.3.8.— Tristeza.

RF-6.3.9.— Sorpresa.

RF-6.3.10.— Confianza.

RF-7.— La categoría de análisis geográfico contendrá la siguiente información:

RF-7.1.— Mapa interactivo con los lugares mencionados en la noticia señalados mediante punteros en el mapa.

RF-8.— La categoría de noticia accesible contendrá toda la información relevante de la noticia dispuesta en un formato apto para ser navegable con Screen Readers y teclado.

RF-9.— La categoría de análisis del texto contendrá el texto de la noticia, así como la opción de resaltar las entidades de dicho texto. Estas entidades están divididas en cuatro categorías: personas, organizaciones, localizaciones y otros.

RF-10.— El usuario deberá poder navegar entre las distintas categorías.

RF-11.— El usuario deberá poder analizar otra noticia una vez rellenado y enviado el primer formulario descrito en el RF-1.1.

3.2.2. Requisitos No Funcionales

RNF-1.— La aplicación deberá ser escalable a nuevas funcionalidades.

RNF-2.— La aplicación debe presentar un alto grado de contraste de colores.

RNF-3.— La aplicación debe estar preparada para usuarios con dicronomatopsia.

RNF-4.— La aplicación deberá ser usable.

3.3. Elección de tecnologías

Para el desarrollo de la aplicación, se han hecho uso de una serie de tecnologías y plataformas. A continuación se listaran todas las tecnologías usadas y se describirá la motivación de uso de las mismas.

3.3.1. Tecnologías Frontend

Angular 8

Angular es una plataforma y framework de desarrollo de Single Page Application (SPA) cuya arquitectura está basada en componentes web. Se ha decidido trabajar bajo este framework para el desarrollo del cliente de la aplicación debido a su gran uso por parte de la comunidad, lo que radica en una extensa documentación [8] [9] [10] [11], y por las numerosas facilidades que provee en el desarrollo de aplicaciones destacando su gran integración con tecnologías como RxJs [12] o Angular Material [13] [14].

Además, Angular posee un sistema de templates reactivos que dan dinamismo a la pagina web, integración con herramientas de testing como Jasmine [15] e integración de librerías para comunicarse a través del protocolo HTTP.

Angular Material

Angular Material es un módulo de componentes que adaptan las directivas Material Design [16] de Google a componentes Angular. Se decidió hacer uso de este módulo por su fácil implementación en el proyecto, su carácter reutilizable y su alto nivel de accesibilidad.

Bootstrap 4

Bootstrap [17] es una biblioteca multiplataforma de código abierto para diseño de aplicaciones web. Su extensa documentación, reusabilidad y fácil implementación lo hacen casi indispensable en cualquier aplicación dirigida al público.

Highchart

Highcharts [18] es una biblioteca de software para gráficos escritos en JavaScript. Permite insertar gráficos de muchos tipos en la aplicación con apenas configuración. En concreto, en este proyecto se ha usado el módulo n2chart, que corresponde con la adaptación de esta biblioteca a componentes Angular.

Leaflet

Leaflet [19] es una biblioteca JavaScript de código abierto que sirve para crear aplicaciones que hagan uso de mapas interactivos. En concreto, en este proyecto se ha usado su versión adaptada al framework Angular, llamada ngx-leaflet.

La elección de esta librería frente a otras más comunes como puedan ser Google Maps viene dada por varios motivos. El principal motivo es económico, puesto que Leaflet es gratuito y de código abierto mientras que Google Maps es de pago. Otro motivo que, aunque de menor peso, también hay que considerar, es la capacidad de personalización que nos ofrece Leaflet permitiéndonos elegir el mapa que queremos mostrar, pudiendo incluso dar elección al usuario para elegir entre varios tipos. Estos mapas son completamente personalizables por lo que podemos agregar alguno propio desarrollado para la aplicación o usar algunos ya creados y gratuitos como es Open Street View Map.

RxJS

RxJS es una librería escrita en JavaScript que hace uso de Observadores para obtener una programación reactiva, permitiendo crear aplicaciones que realicen llamadas asíncronas de una manera sencilla. Esto ha permitido crear un servicio alojado en el cliente que se encargue de la comunicación con el servidor haciendo al cliente completamente transparente e independiente.

Jasmine y Karma

Jasmine es un framework de testing que permite crear tests y da soporte al método de desarrollo Behaviour-Driven Development (BDD). Karma [20] es una herramienta que nos permite automatizar la ejecución de tests y tiene buena integración con frameworks como Jasmine.

Estas herramientas han sido escogidas para realizar las pruebas unitarias de esta aplicación debido a su buena integración con Angular y abundante documentación [21].

3.3.2. Tecnologías Backend

Flask

Flask [22] es un micro framework basado en Python diseñado para poner en funcionamiento un servidor de manera sencilla y rápida con capacidad de escalar sin muchos problemas. En concreto, en este proyecto se ha usado Flask Restful [23], el cual es una variante de Flask que proporciona herramientas para montar un servidor API REST.

La elección de la tecnología a implementar en el servidor fue complicada debido a la gran cantidad de opciones disponibles. Se optó por Flask Restful porque que se ajustaba perfectamente a la idea inicial propuesta durante la fase de análisis del proyecto, por su fácil puesta en marcha y por bajo consumo de recursos.

Además, dentro de este framework se encuentran otras herramientas útiles de las cuales se ha hecho uso como son Flask Cache [24], el cual provee al servidor de un servicio simple de cache para mejorar el rendimiento de las peticiones, y Flask CORS, el cual dota a todas las comunicaciones HTTP que realice el servidor de cabeceras adicionales para garantizar mayor seguridad siguiendo el mecanismo de intercambio de recursos de origen cruzado.

Newspaper3k

Newspaper3k [25] es una librería Python que dota al programador de herramientas para analizar y procesar noticias. Con ella pueden extraerse datos básicos de la noticia como son autores, fecha de publicación o la imagen principal pero también permite realizar un procesamiento de la misma para obtener palabras clave y un resumen.

El motivo de uso de esta librería es bastante evidente pues obtiene, de manera sencilla y sin apenas configuración, mucha información básica sobre la noticia de la que la aplicación hace uso. Además, cuenta con soporte para varios idiomas además del español, haciéndolo un candidato ideal para futuras ampliaciones de la aplicación a otros idiomas.

SpaCy

SpaCy [26] es una librería de código abierto basada en Python para el procesamiento de lenguaje natural. Esta librería cuenta con una vasta variedad de herramientas como lematizadores, tokenizadores o reconocimiento de entidades entre otras cosas. Cuenta con soporte para varios idiomas y una gran cantidad de documentación. El proyecto hace uso de esta librería en numerosas categorías como la de análisis del texto, análisis geográfico o análisis emocional.

Geopy

Geopy [27] es una librería Python especializada en la geocodificación de direcciones, nombres de localidades, países o lugares. En este proyecto se hace uso de esta librería en conjunción con SpaCy para obtener las coordenadas de los lugares mencionados en el texto.

Textblob

Textblob [28] es una librería Python especializada en el procesamiento de lenguaje natural. Esta herramienta permite realizar un análisis de sentimientos, extracción de frases nominales o clasificación entre otras muchas cosas. Para este proyecto se ha hecho uso de su herramienta de análisis de sentimientos capaz de calcular un porcentaje de polaridad y subjetividad teniendo en cuenta la intensidad de las palabras.

EmoLex

EmoLex [29] se trata de una colección de diccionarios de palabras que asocian cada una de estas a una o varias emociones de un grupo de doce. Este diccionario fue construido por el National Research Council of Canada en 2016. Esta colección fue desarrollada en inglés y posteriormente traducida al español mediante Google Traductor.

El uso de este lexicón en contrapartida con otras herramientas mejor preparadas para el idioma español se debe a que estas otras herramientas eran en su mayoría de pago y/o solo permitían diferenciar entre sentimiento positivo o negativo mientras que este lexicón provee de un mayor grado de granularidad.

3.4. Plataforma y herramientas de trabajo

3.4.1. Plataforma de trabajo

Ubuntu

Ubuntu es un sistema operativo de código abierto y libre. Es una distribución Linux basada en Debian. El proyecto ha sido desarrollado en este sistema operativo por su libertad a la hora de instalar nuevas tecnologías y porque la otra opción en consideración era Windows, y esta daba conflicto con alguna de las tecnologías usadas.

Visual Studio Code

Visual Studio Code [30] es un editor de código fuente avanzado. Incluye numerosas funcionalidades que ayudan y simplifican el trabajo del programador como soporte para la depuración, control integrado de Git, resaltado de sintaxis para distintos lenguajes, finalización inteligente de código, fragmentos y refactorización de código e integración con una gran librería de extensiones que amplían aun más su funcionalidad.

3.4.2. Herramientas de trabajo

GitHub

GitHub [31] es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git. Git a su vez es un gestor de versiones de código abierto que permite almacenar un historial de cambios realizados al proyecto, revertir dichos cambios y dividir y/o fusionar el proyecto en ramas de desarrollo paralelas.

NPM

NPM [32] es un sistema de gestión de paquetes para Node.js. Proporciona un cliente para la línea de comandos que permite instalar y gestionar las versiones de los distintos paquetes instalados para nuestro proyecto. Además, cuenta con una biblioteca abierta de paquetes para desarrollo frontend llamada NPM Registry.

Para gestionar las versiones de nuestros paquetes, NPM almacena estos datos en un fichero llamado `package.json`, el cual incluye el nombre y versión de cada paquete instalado, así como una distinción entre paquetes que usa la aplicación y paquetes que son solo necesarios para el desarrollo de la misma. Además, permite llevar un control de la versión de nuestra propia aplicación.

Balsamiq

Balsamiq es un programa desarrollado por Balsamiq Studios que permite la creación de maquetas de una manera sencilla e intuitiva.

Trello

Trello [33] es un software de administración de proyectos. Permite crear tableros con distintas columnas y cartas en cada columna. Estas cartas pueden contener información sobre tareas como son un título, descripción, personas involucradas y fechas límite. Trello es especialmente útil durante desarrollos que sigan metodologías ágiles como Scrum o desarrollos incremental iterativos. En este proyecto, se ha hecho uso de la herramienta para llevar un control del Product Backlog.

PlantUML

PlantUML [34] es una herramienta de código abierto que permite crear diagramas UML a partir de un lenguaje de marcado. Se ha utilizado para el desarrollo de algunos diagramas presentes en la documentación.

3.5. Metodología de trabajo

Para el desarrollo de este proyecto se ha seguido una metodología de trabajo incremental iterativa [35]. Este tipo de metodología consiste en la división de la planificación del proyecto en bloques temporales de 2 a 4 semanas llamados iteraciones.

En cada iteración se repiten todas las fases de desarrollo de un proyecto desde la fase de análisis hasta la de integración y pruebas. Durante cada iteración se desarrollan ciertos requisitos del proyecto hasta obtener un producto final entregable al cliente. Después, durante la siguiente iteración, partiendo del producto entregado, se añade funcionalidad.

De esta manera, el proyecto goza de gran flexibilidad a la hora de modificar requisitos, permite al cliente obtener un producto rentable desde un tiempo relativamente temprano y permite probar el core de la aplicación de manera extensa puesto que es lo primero que se desarrolla y se va probando e integrando con la aplicación durante cada iteración.

En concreto, para este proyecto se ha usado una adaptación de esta metodología compuesta de 5 fases. En la tabla 3.1 podemos ver un resumen de dichas fases y su duración con respecto a los 8 meses de duración que tuvo el desarrollo de este proyecto.

La primera fase consistió en un estudio previo del estado del arte, viabilidad del proyecto y tecnologías disponibles en el mercado.

Fases	Meses	Iteración	Requisitos
Fase 1	Meses 1 a 3	Estudio previo	
Fase 2	Meses 4 y 5	Iteración 1	RF1-RF3,RF4.1, RF5, RF10-RF11,RNF1
Fase 3	Mes 6	Iteración 2	RF4.2-RF4.3, RF6-RF7
Fase 4	Mes 7	Iteración 3	RF4.4-RF4.5,RF8-RF9, RNF2
Fase 5	Meses 7 y 8	Pruebas con usuarios y memoria	

Tabla 3.1: Tabla de planificación del proyecto

Durante la segunda fase se desarrolló la primera iteración del proyecto la cual engloba la puesta en funcionamiento del cliente y el servidor, la conexión entre ambos y el desarrollo de la categoría de Datos Básicos.

Durante la tercera fase se desarrolló la segunda iteración la cual engloba las categorías de Análisis Geográfico y Análisis Emocional.

Durante la cuarta fase se desarrolló la tercera y ultima iteración que engloba las categorías de Noticia Accesible y Análisis del Texto, así como la adición de un modo de alto contraste y la definición de una paleta de colores consistente para la aplicación.

Por último, en la quinta fase se realizaron pruebas con usuarios y se escribió este documento.

Durante todo el desarrollo del proyecto, se siguió un plan de acción reflejado en el product backlog del Anexo B donde pueden verse la evolución de las tareas en las distintas iteraciones.

DISEÑO Y DESARROLLO

En este capítulo se detallarán todos los aspectos referentes a las fases de diseño y desarrollo del proyecto. Esto incluye la definición de la arquitectura de la aplicación, las maquetas, detalles de implementación durante la fase de desarrollo y un apartado sobre la accesibilidad en la aplicación.

4.1. Arquitectura de la aplicación

La aplicación sigue una estructura cliente-servidor bastante estándar. El cliente es el plugin, el cual está desarrollado en el framework Angular 8 y se encarga de toda la funcionalidad del mismo. El servidor se trata de una API REST desarrollada en Flask que se encarga de todo el trabajo de análisis y extracción de datos de la noticia. Debido a la falta de necesidad de guardar información, la aplicación no cuenta con base de datos.

4.1.1. Frontend

El framework Angular, desde su segunda versión en adelante, trabaja con una arquitectura basada en componentes. Esta arquitectura consiste en dividir la funcionalidad en componentes auto-contenidos reutilizables que agrupen parte del modelo de la aplicación, la vista relacionada y el controlador correspondiente.

Aunque esta es la explicación más sencilla, Angular no sigue un patrón Model-View-Controller (MVC) [36] [37]. Eso se debe principalmente a su naturaleza reactiva y a la herramienta de two-way binding que posee. Two-way binding es una herramienta que permite que los cambios en la vista afecten al modelo, cosa que contradice uno de los principios del patrón MVC tradicional. Un patrón más apropiado, pero que no casa completamente, es Model-View-ViewModel (MVVM). Este patrón se compone de un modelo, una vista y luego un tercer agente abstracto llamado ViewModel. Este agente, el cual es una parte de la vista, se encarga de la relación entre el modelo y la vista de manera que si algo cambia en uno, se refleja dicho cambio en el otro.

En el Anexo C podemos ver una representación esquematizada de la organización de los com-

ponentes en la aplicación. Esta se divide en 3 módulos: App Module, el cual se encarga de iniciar la aplicación y contiene los componentes básicos de la misma; Category Module, el cual reúne todos los componentes y funcionalidad relacionadas con las categorías de la aplicación; y Routing Module, el cual se encarga de el enrutado de todos los componentes.

Comenzando con el más básico, Routing Module contiene un solo componente:

App Routing Component: Componente que se encarga de enrutar todos los componentes de la aplicación.

A continuación describimos App Module, el cual contiene 2 componentes:

Welcome Component: se encarga de administrar todo lo referente a la página de inicio de la aplicación. En ella se encuentra el formulario para que el usuario introduzca (o seleccione con el botón) la URL que desea analizar. Este componente también se encarga de enviar la URL a analizar a la API haciendo uso del servicio backend para que valide dicha URL.

App Component: se encarga de poner en funcionamiento la aplicación. Por convención, no tiene mayor funcionalidad que esa.

Por ultimo, vemos el módulo más complejo que guarda la mayoría de la funcionalidad del plugin, llamado Category Module. Este módulo se compone de 8 componentes descritos a continuación:

Categories Component: Componente que se encarga de administrar todo lo referente a la página de selección de categorías. Desde ella se puede acceder a todas las categorías y contiene instancias de estas.

Basic Data Component: Componente que se encarga de administrar todo lo referente a la categoría de datos básicos de la página web. Dichos datos son obtenidos a través del servicio backend que a su vez llama a la API para que obtenga los datos de la URL guardada. Este componente implementa la interfaz de ICategory y hace uso de la interfaz IBasicData para almacenar los datos obtenidos por el servidor.

Emotion Component: Componente que se encarga de administrar todo lo referente a la categoría de Análisis Emocional de la página web. En ella se muestran gráficas y datos referentes a las distintas mediciones realizadas al texto de la noticia donde se representa la carga emocional del texto incluyendo, subjetividad, polaridad y grado de presencia de ciertas emociones. Dichos datos son obtenidos a través del servicio backend que a su vez llama a la API para que obtenga los datos de la URL guardada. Dentro de este componente se usan los componentes son SolidGaugeChart y SpiderwebChart para representar los diagramas de la categoría. Este componente implementa la interfaz de ICategory.

SolidGaugeChart y SpiderwebChart Component: Componentes que se encargan de representar gráficos. Son reutilizables y sus datos vienen dados por unos parámetros de

entrada necesarios para su instanciación.

Geografic Component: Componente que se encarga de administrar todo lo referente a la categoría de Análisis Geográfico de la página web. En ella se un mapa interactivo con las localizaciones señaladas, si las hubiese, mencionadas en el texto de la noticia. Dichos datos son obtenidos a través del servicio backend que a su vez llama a la API para que obtenga los datos de la URL guardada. Este componente implementa la interfaz de ICategory.

Accesibility Component: Componente que se encarga de administrar todo lo referente a la categoría de Noticia Accesible. En ella se representa la información básica de la noticia, como son autores, fecha de publicación y el texto, con un HTML accesible que permite la navegación por teclado y el uso de Screen readers.

Text Analisis Component: Componente que se encarga de administrar todo lo referente a la categoría de Análisis del Texto. En ella se muestra el texto de la noticia con las entidades de dicha noticia resaltadas.

Además de los tres módulos principales del plugin, existe un cuarto agente que es independiente de los otros módulos. Este es el Backend Service, el cual es el servicio encargado de comunicarse con el servidor backend y hacer de middleware para todas las peticiones HTTP que realice la aplicación a dicho servidor. Además, guarda variables compartidas entre todos los componentes como es la variable del modo de alto contraste.

4.1.2. Backend

Como se ha mencionado anteriormente, el servidor backend esta desarrollado en el microframework Flask. En concreto, se está usando una variante de Flask llamada Flask Restful pensada para montar servidores API REST. Un servidor Restful es aquel que provee al cliente de una interfaz uniforme a través de HTTP para acceder a sus recursos. REST define una metodología en la cual el servidor no guarda el estado y cada petición es independiente de la anterior y permite interactuar con el servidor a través de peticiones HTTP GET, POST, PUT y DELETE.

El servidor desarrollado en este proyecto se compone de 4 clases que dividen la funcionalidad en grupos semánticos y jerarquiza la interfaz de entrada. Estas cuatro clases son:

AnalyzeUrl: se encarga de validar una URL enviada y ver si es compatible con lo que acepta el servidor. Actualmente el servidor solo acepta URL de noticias. Para acceder a esta funcionalidad se debe enviar una petición POST con la URL a la dirección URI + /analiceUrl.

BasicData: se encarga extraer la información básica de la noticia y devolverla en formato JSON. Para acceder a esta funcionalidad se debe enviar una petición POST con la URL a la dirección URI + /basicInfo/<option> donde option puede ser:

- all
- title
- authors
- publishDate
- keywords
- summary
- text
- topImg
- movies

Spacy: se encarga de extraer información del lenguaje del texto usando la librería Spacy y devolverla en formato JSON. Para acceder a esta funcionalidad se debe enviar una petición POST con la URL a la dirección URI + /spacy/<option> donde option puede ser:

- all
- locations
- organizations
- persons

Emotion: se encarga de extraer información relacionada con emociones del texto de la noticia y devolverla en formato JSON. En concreto, extrae la polaridad, subjetividad y el grado de presencia de cada emoción de una lista de 12 emociones. Para acceder a esta funcionalidad se debe enviar una petición POST con la URL a la dirección URI + /emotion.

4.1.3. Comunicación entre cliente y servidor

La manera en la que se entrelazan cliente y servidor y los distintos componentes de cada parte se representan de manera generalizada en el diagrama de secuencia de la figura 4.1. En esta podemos ver que cuando un componente requiere de un dato, se lo pide al servicio backend y es este el que envía una petición HTTP al servidor. El componente crea un Observable que se suscribe a al servidor a la espera de que este reciba una respuesta. Cuando el servidor termina de analizar la petición, devuelve el resultado al servicio y es este quien notifica al componente de que ha recibido los datos.

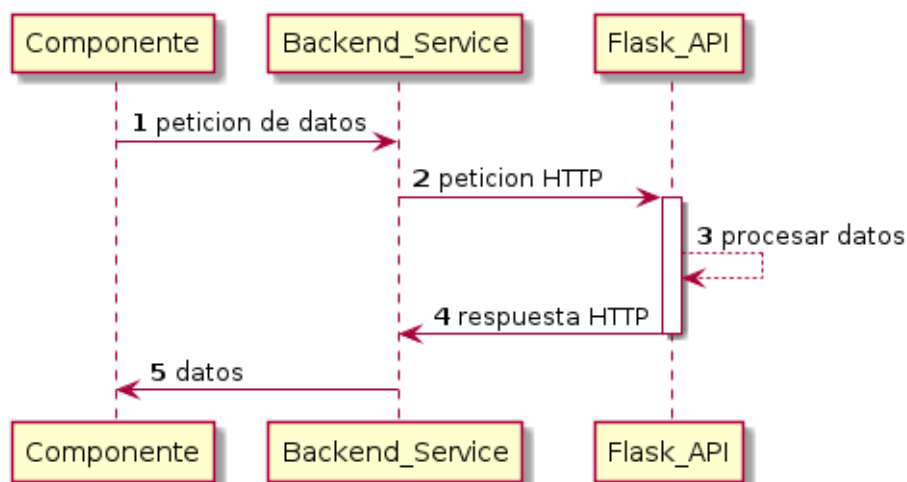


Figura 4.1: Generalización de la comunicación entre cliente y servidor

4.2. Maquetas

En el Anexo D se muestran las maquetas desarrolladas durante la fase de diseño de las distintas iteraciones. En estas maquetas se muestran las distintas pantallas a las que puede acceder el usuario. Cabe destacar que varían ligeramente del resultado real (Véase Anexo E) por la limitación de la herramienta usada a la hora de crear maquetas y reajustes menores de diseño durante el desarrollo. Aun así, la idea conceptual de cada maqueta sigue estando presente en la aplicación.

4.3. Desarrollo del proyecto

4.3.1. Detalles de implementación del cliente

Durante el desarrollo del cliente, se han tenido que tomar ciertas decisiones de implementación que cabe mencionar.

Siguiendo un orden cronológico de desarrollo la primera decisión que hubo que tomar durante el desarrollo tuvo que ver con la pantalla de selección de categorías. El componente de categorías debía conocer las categorías existentes para poder mostrarlas por lo que se optó por guardar una copia instanciada de cada categoría dentro del componente como podemos ver el extracto de código 4.1. De esta manera, el componente podría acceder a los datos de cada categoría y mostrar título, descripción y el icono correspondiente sin necesidad de controlar esa parte de la funcionalidad.

```
1
2 categories: ICategory[] = [
3   new BasicDataCategoryComponent (this.backendService),
4   new LocationCategoryComponent (this.backendService),
5   new EmotionCategoryComponent (this.backendService),
6   new AccesibilityCategoryComponent (this.backendService),
7   new TextAnalysisCategoryComponent (this.backendService)
8 ];
```

Código 4.1: Extracto de código del componente Categories

Esta solución, aunque funcional, dio algunos problemas. Empezando por que al agregar una nueva categoría, se debe manualmente modificar el *categories* y agregarlo a la lista de categorías. Otro problema es que toda funcionalidad que se ejecute al instanciar el componente de cada categoría, se ejecutará dos veces pues se instancia ahí y al cargar el componente cuando el usuario entra dentro de una categoría.

Se barajaron otras opciones como la de que el componente de categories almacenase solo los datos de cada categoría, en vez de la categoria entera, pero esto choca un poco con el diseño de un componente en el cual todo lo relacionado con dicho componente esta autocontenido.

Otra decisión de implementación que hubo que tomar fue dónde almacenar la variable de modo de alto contraste de manera que todos los componentes pudiesen acceder a dicha variable. La primera opción que se barajó fue usar el patrón Redux [38] para almacenar el estado de la aplicación. Redux es una librería JS que maneja el estado de la aplicación donde se pueden almacenar distintos datos que competen a toda la aplicación. En Angular, existe una implementación de este patrón llamado ngRx.

El patrón Redux se compone de tres conceptos y tres principios básicos. Los conceptos son los siguientes:

Store: Es el objeto que contiene el estado de la aplicación.

Acciones: Son un bloque de información que envía datos desde tu aplicación al store. Son la única fuente de información para el store.

Reducer: Es el encargado de actualizar la información del store. En vez de modificarlo, crea una copia y lo sustituye. Es importante destacar que un reducer es siempre una función pura.

Los tres principios básicos en los que se basa el patrón Redux son:

Única fuente de verdad: El estado de toda tu aplicación esta almacenado en un único store.

El estado es solo de lectura: La única forma de modificar el estado es emitiendo una acción, un objeto que describe qué ocurrió.

Los cambios se realizan con funciones puras: Para especificar como el árbol de estado es

transformado por las acciones, se utilizan reducers puros.

Esta opción, que en la teoría es bastante acertada a lo que necesitábamos, es bastante complicada de implementar y, teniendo en cuenta que nuestro estado de la aplicación es muy simple, se decidió realizar una adaptación de dicho patrón de manera simplificada usando el servicio que se comunica con el backend. Dicho servicio, como ya hemos visto, es accesible desde todos los componentes por lo que se creó una variable y se almacenó allí. Además, se crearon métodos para modificarlo y que notifique la modificación a los distintos componentes. Podemos ver la implementación final en el extracto de código 4.2. Esta solución es sencilla y simple de implementar pero poco escalable si el estado de la aplicación crece.

```
1 private messageSource = new BehaviorSubject(false);
2 showDaltonicMode = this.messageSource.asObservable();
3
4
5 changeDaltonicMode(message: boolean) {
6     this.messageSource.next(message);
7 }
```

Código 4.2: Extracto de código del servicio Backend

Una tercera decisión de diseño que cabe destacar fue la de crear un patrón de colores en la categoría de análisis emocional. Para esta categoría, se pretendía representar la mayor cantidad de información posible de manera visual sin que hubiese que acceder a distintos menús o hacer uso de botones. Por ello, y por el limitado espacio del que se dispone, se buscó una manera de indicar el significado de los gráficos sin necesidad de texto.

Para ello se ideó un sistema de cinco colores que van desde el verde hasta el rojo haciendo uso de la asociación común del verde con bueno y rojo con malo. De esta manera, los gráficos de radar representados en la categoría, a medida que aumentan o disminuyen su valor, cambian de color. Para una mayor claridad para el usuario, se incluyó una columna donde se ve toda la paleta de colores representada con símbolos de caras que van desde la expresión contenta hasta la enfadada pasando por la indiferencia. Esta paleta fue adaptada a una escala de grises para el modo de alto contraste [39].

Con respecto al servidor, también se tomaron algunas decisiones que podemos mencionar. La primera, fue la idea de realizar un preprocesado del texto antes de analizarlo para minimizar el error. En este preprocesado, se incluye lematizar las palabras y eliminar las stopwords. Para lematizar la entrada se usó la librería de SpaCy y las stopwords se sacaron de un lexicon [40].

En el servidor se hace uso de una cache. Esta cache es provista por Flask y consiste en un diccionario simple de clave-valor. Si bien no es el sistema más eficiente, y Flask te pone a disposición otros sistemas como memcache, estos requieren de un mayor desarrollo o de otro servidor reservado para la cache por lo que se optó por ir a la solución más sencilla y funcional. Aun así, el uso de esta cache se ha notado en los tiempos de carga cuando se revisan noticias que hemos analizado anteriormente.

por último cabe destacar que toda la aplicación esta preparada para funcionar única y exclusivamente en español. La adaptación a otros idiomas no sería complicada pero requeriría de un nuevo desarrollo y una nueva fase de pruebas específicas.

4.4. Accesibilidad

Cuando hablamos de accesibilidad web nos referimos al desarrollo de aplicaciones web que sean usables por la mayor cantidad de personas. De esta manera, no solo nos estamos refiriendo a un público que hable diferentes idiomas sino que pueda presentar también algún tipo de condición que impida su navegación por una web tradicional (p.e. una persona con dificultad visual o auditiva). Para que una aplicaciones web sea accesible, deben de ofrecer igual acceso y oportunidades a todo el mundo.

Durante los últimos años ha habido una gran revolución en este sector y es cada vez mayor el número de empresas que desarrollan pensando en la accesibilidad. La World Wide Web Consortium (W3C), una comunidad internacional que desarrolla estándares para asegurar el buen crecimiento a largo plazo de la Web [41], desarrolló la Web Content Accessibility Guidelines (WCAG), la cual define una serie de técnicas, guías y especificaciones que describen soluciones accesibles [42] a problemas y situaciones comunes durante el desarrollo de aplicaciones web.

Por esto, la accesibilidad ha sido un factor clave durante el desarrollo de esta proyecto. Desde su prima fase inicial se definieron medidas y requisitos que asegurasen que la aplicación fuese accesible en cierto nivel. Aún hay espacio para el crecimiento en esta materia, y hay bastantes cosas que mejorar, pero alguna de las soluciones implementadas conviene destacarlas.

La primera y más transversal, fue seguir las indicaciones propuestas por la WACG junto con algunas indicaciones de Google Material. Esto incluye la codificación de un buen HTML, haciendo un uso correcto de las etiquetas HTML5 como son main, article o header, pero también incluye el seguimiento del standard ARIA.

Las etiquetas ARIA pretenden otorgar de significado semántico a elementos HTML que no lo tienen. De esta manera, puedes hacer uso de elementos que contengan etiquetas div, las cuales carecen de significado, sin que esto perjudique a los screen readers, que no saben como tratar estas etiquetas. Esta medida mejora enormemente la navegación mientras se esta haciendo uso de alguna de estas herramientas.

Otra medida que se tomó fue la inclusión de un modo de alto contraste basado en una escala monocromática de grises. Esto mejora la lectura para aquellos que tengan una visibilidad reducida pero también ayuda a aquellos que padezcan algún tipo de daltonismo, sobre todo en la categoría de Análisis Emocional, la cual depende enormemente de los colores para presentar la información.

PRUEBAS Y RESULTADOS

En este capítulo se detallarán todos los aspectos referentes a la fase de pruebas del proyecto. Se definirán las pruebas y los métodos seguidos y se presentarán los resultados finales de los datos obtenidos por las pruebas con usuarios.

5.1. Pruebas Unitarias

Durante la fase de pruebas de cada iteración, se desarrollaron pruebas unitarias para cada componente. En estas pruebas se hizo uso de Jasmine para el desarrollo de las mismas y Karma para la ejecución de estas de manera automatizada. Jasmine sigue una estructura específica para desarrollar los test denominada Arrange, Act, Assert (AAA). Esta estructura de pruebas divide los test en 3 segmentos. En el primer segmento, Arrange, se instancian todos los datos que luego serán utilizados en las pruebas. En el segundo segmento, Act, se ejecuta la función o funciones a probar. En el último segmento, Assert, se comprueban los resultados obtenidos en el Act y se comparan con los resultados esperados.

Esta estructura lleva consigo asociada otro término dominado DRY. En el desarrollo de código existen dos principios para referirse a la manera de escribir código, DUMP, Descriptive And Meaningful Phrases, y DRY, Do not Repeat Yourself. DUMP, promueve la legibilidad del código haciendo algunos pasos más explícitos a costa de repetir o no reducir tanto el código. DRY promueve un código limpio y reducido en el que no haya extractos de código duplicados.

En la fase de pruebas, los test desarrollados deben llegar a un balance entre estas dos partes. Una asegura que el código sea mantenible y fácilmente legible por cualquier persona mientras que la otra, al reducir el código a su mínima expresión, evita repetir código durante las pruebas, lo que conlleva a que los test sean más específicos y prueben sólo la funcionalidad que se quiera examinar.

Jasmine tiene 4 métodos principales:

Describe: En este método se listan toda la batería de test a realizar.

BeforeEach: En este método se instancian todas las variables y configuraciones necesarias

para correr los test. Este método corresponde al segmento Arrange de AAA.

It: En este método se encapsula la lógica del programa que se quiere probar. Este método corresponde al segmento Act de AAA.

Expect: En este método se pone la afirmación que el resultado de it() debe cumplir. Para ello existen funciones auxiliares como toBe(), toContain(), toHaveBeenCalled() o toHaveBeenCalledWith().

En el extracto de código 5.1 podemos ver un extracto del test unitario realizado al componente Welcome.

```
1 describe('WelcomeComponent', () => {
2   let component: WelcomeComponent;
3   let URL: string;
4   let PostError: boolean;
5   let PostErrorMessage: string;
6   let mockBackendService;
7   let mockRouter;
8   let response;
9
10  beforeEach(() => {
11    URL = "http://www.google.es";
12    PostError = false;
13    PostErrorMessage = "";
14
15    mockBackendService = jasmine.createSpyObj(["postAnaliceUrl"]);
16    mockRouter = jasmine.createSpyObj(["navigate"]);
17
18    component = new WelcomeComponent(mockBackendService, mockRouter);
19  });
20
21  describe("buttonClick", () => {
22    it("should call postAnaliceUrl with correct url", () => {
23      component.url = URL;
24      mockBackendService.postAnaliceUrl.and.returnValue(of(true));
25
26      component.buttonClick(URL);
27
28      expect(mockBackendService.postAnaliceUrl).toHaveBeenCalledWith(URL);
29    });
30
31    it("should call router.navigate if url is correct", () => {
32      response = URL;
33      mockBackendService.postAnaliceUrl.and.returnValue(of(response));
34
35      component.buttonClick(URL);
36
37      expect(mockRouter.navigate).toHaveBeenCalled();
38    });
39  });
40 });
```

Código 5.1: Extracto de código del test del Welcome Component

En esta pieza de código podemos ver también el uso de un método de Jasmine llamado `createSpyObj()` que es almacenado en la variable `mockBackendService`. Mock es un término que se usa en testing para referirse a un objeto que se hace pasar por otra cosa. En este caso, este mock está simulando el servicio de backend real. Esto se usa en los test unitarios cuando una clase depende de agentes externos y para poder probarla, necesitas que estos agentes externos existan. De esta manera puedes crear un objeto sustituto que simule un servicio backend pero sin tener que usar el real pues este puede contener también errores que diluirían el objetivo del test unitario.

Jasmine proporciona algunos métodos para crear distintos tipos de mocks. Existen mock que cumplen simplemente una interfaz y no tienen funcionalidad interna, los hay espías que recogen la entrada y la comprueban con lo que se supone que debería entrar y devuelven una salida correcta y los hay que contienen dentro del mock una funcionalidad mínima si el test lo requiere, por ejemplo, llamar a una API falsa controlada que devuelva datos falsos.

Después de la codificación de los test con Jasmine, se usó Karma para ejecutar todos los test del proyecto de manera automatizada. Karma ha de ser configurada a través de un archivo JSON para indicar cuál es la extensión de los archivos de test y qué tecnología usan, en esta caso Jasmine.

Como podemos observar en la figura 5.1, la herramienta crea una página web donde muestra los resultados de cada grupo de test, el número de tests realizados y el feedback de cada test fallido. En la figura 5.2 podemos ver un ejemplo de batería de test con todas las pruebas pasadas correctamente.

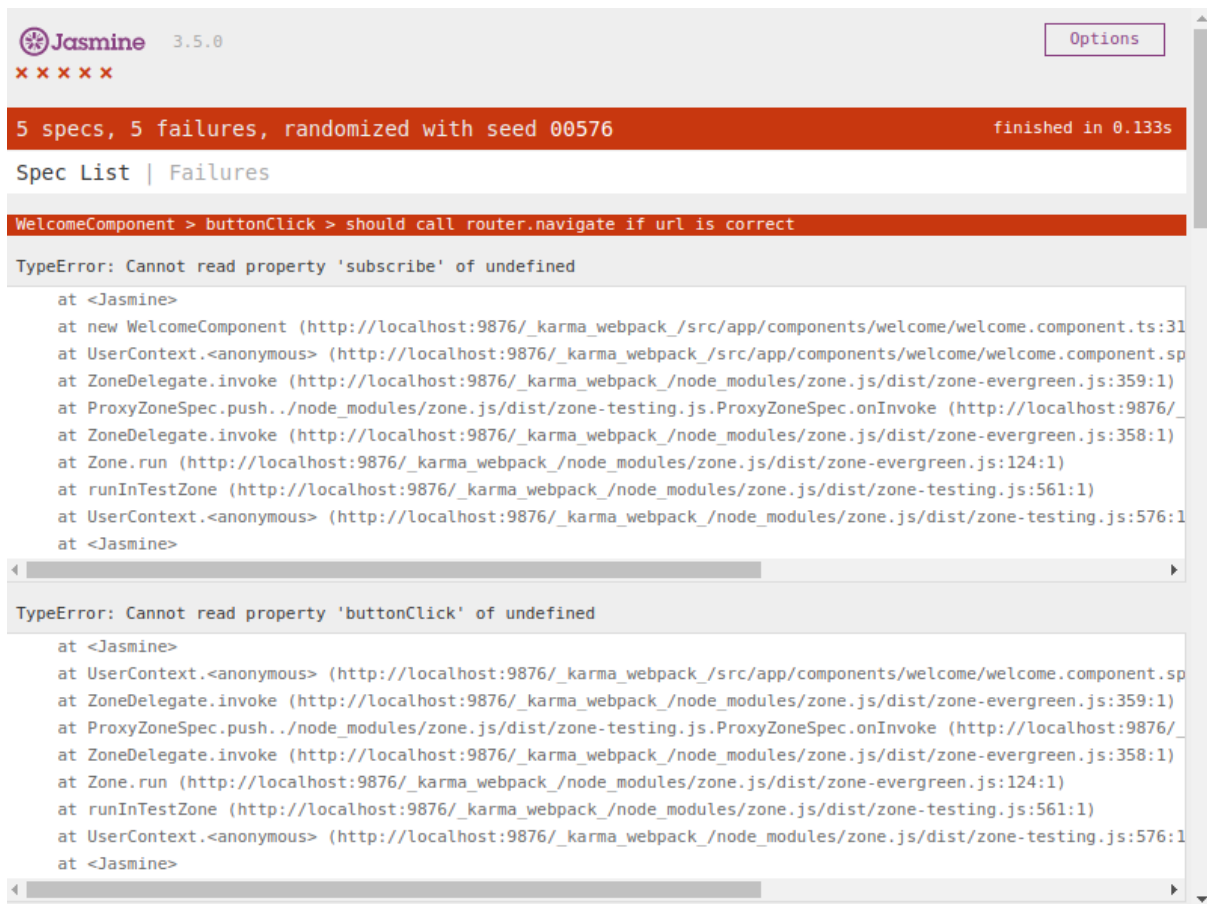


Figura 5.1: Detalles del test unitario fallido del componente Welcome



Figura 5.2: Resumen del test unitario sin fallos del componente Welcome

5.2. Pruebas con usuarios

Tras acabar las tres iteraciones de desarrollo planeadas, se realizó una cuarta fase consistente exclusivamente de pruebas con usuarios. Durante esta fase de pruebas, se han realizado test de usabilidad a 13 usuarios, estando cada test dividido en dos partes. La primera parte consiste en un ejercicio guiado con una serie de pasos que debe realizar el usuario mientras navega por la aplicación. Antes de comenzar la prueba, se le pide al usuario que lea y rellene un informe de consentimiento de datos (véase Anexo F). El ejercicio comienza con una noticia que se le presenta al usuario y se le pide lo siguiente:

- Lea la noticia en la que se encuentra.
- Utilizando el plugin, analice la noticia y encuentre la categoría donde se muestran la fecha de publicación y los autores de la noticia.
- Vuelve a la pantalla de selección de categorías y encuentra la categoría donde se muestren los lugares que han sido mencionados en la noticia.
- Vuelva a la pantalla de selección de categorías y busque la categoría en la que pueda realizar un análisis de las emociones y describa que representa cada gráfico.
- Vuelva a la pantalla de selección de categorías y busque la categoría donde se realice un análisis del texto de la noticia, encuentre donde puede resaltar las entidades presentes en el texto y describa qué tipo de entidades existen en esta noticia.
- Vuelva a la página de inicio, active el modo de alto contraste y revise la categoría de análisis emocional.
- Ahora se le va a presentar una nueva noticia, léala, analice la noticia con el plugin y navegue por él libremente.

Durante la realización de estos pasos, se le pide al usuario que realice juicios de valor y crítica sobre la aplicación, y se anotan todas los errores o posibles mejoras en una lista.

Esta serie de pasos han sido escogidos por varias razones. La primera es que, con estos pasos, se le obliga al usuario a navegar por todas las categorías relevantes. Además, se pone a prueba la intuitividad de los títulos, descripciones y símbolos de cada categoría.

Otra razón es que, durante los pasos 4 y 5, se pide al usuario que describa la información presentada en la categoría para poder analizar si el usuario está entendiendo realmente el significado de cada gráfico o de las abreviaturas de las entidades. Con el paso 6, se comprueba además si el código de colores presentado en la categoría de análisis emocional sigue siendo válido con el modo de alto contraste activado. Por último, en el paso final, se presenta al usuario otra noticia con distinto nivel de subjetividad y polaridad y pido que hagan uso del plugin para que puedan comparar información y juzgar la utilidad del plugin.

Durante las pruebas, se dividieron a los usuarios en dos grupos. El primer grupo realizaría los pasos indicados anteriormente pero el segundo grupo realizaría una variación de estos. Esta variación consiste en intercambiar los pasos 4 y 6. El motivo de esta variación es comprobar de manera más exhaustiva si el código de colores de la categoría de análisis emocional sigue teniendo sentido con el modo de alto contraste activado. La explicación de por qué este cambio es importante, es que durante la realización de los pasos sin variaciones, el usuario visita primero la categoría con los colores normales por lo que realiza ya una asociación de colores, símbolos y posición de estos mismos. Cuando visita nuevamente la categoría con el modo de alto contraste activado, ya tiene esta asociación en la cabeza por lo que su juicio sobre este modo puede verse comprometido. Intercambiando estos pasos, se asegura que, independientemente de si el modo está o no activado, el código de colores sea juzgado de manera de la misma manera.

Las personas que fueron entrevistadas pertenecen a distintos perfiles y fueron escogidos a propósito. Un grupo de 4 personas pertenecían a un perfil joven con alto manejo de la tecnología y con ciertos conocimientos informáticos. Un segundo grupo de 5 personas pertenecían a un perfil de usuario joven pero sin estos conocimientos informáticos y con un manejo de la tecnología alto pero menos que el grupo anterior. Por último, un grupo de 4 personas pertenecían a un perfil adulto de usuario con un nivel de manejo de la tecnología bajo. Estos perfiles fueron escogidos para aumentar la diversidad de las pruebas, poder obtener opiniones de distintos los rangos de edades y habilidades y evitar así sesgo en los resultados.

Los errores detectados durante esta parte de la prueba fueron los siguientes:

- presente en varias situaciones, existen momentos en los que la aplicación está cargando pero no se muestra ninguna confirmación visual, por lo que el usuario piensa que no esta funcionando.
- En la página de inicio, la funcionalidad del botón Analizar pagina actual no esta clara.
- En la página de selección de categorías, los usuarios no vieron que existían más categorías si hacían scroll vertical.
- En la página de selección de categorías, los usuarios no entendieron el propósito de la categoría Noticia Accesible ni de la categoría Análisis del Texto.
- En la página de selección de categorías, existe un error que se da a veces si el usuario mueve el cursor demasiado rápido, en el que se selecciona una categoría distinta de la que el usuario está señalando.
- El término entidades presente en la categoría de Datos Básicos y Análisis del Texto no se entiende.
- En la categoría de Datos Básicos, en la sección de entidades, algunos de los subgrupos presentan errores con el scroll vertical.

- En la categoría de Análisis del texto, una de las entidades se hace referencia con la abreviación MISC pero en la categoría de Datos Básicos, a esa misma categoría de entidad se le denomina Otros.
- En la categoría de Análisis del texto, el significado de la categoría de cada entidad no es del todo claro.
- En la categoría de Análisis Emocional, los usuarios no comprendieron porque los gráficos de subjetividad y polaridad se encontraban en esa categoría.
- En la categoría de Análisis Emocional, el código de colores no es del todo claro. El distinto uso que se le da en los gráficos de polaridad y subjetividad confundió a algunos usuarios.
- En la categoría de Análisis Emocional, los usuarios no entendían que representaba el gráfico de Polaridad.
- En la categoría de Análisis Emocional, existe un error con el menú de opciones presente en cada gráfico que bloquea la aplicación.
- En la categoría de Noticia Accesible, existe un error con el botón de Atrás.

Más adelante se entrará en detalle analizando el posible motivo de cada fallo proponiendo una solución acorde.

La segunda fase de las pruebas consiste en un cuestionario que el usuario debe rellenar que consta de varias afirmaciones y una puntuación del 1 a 5, siendo 1 totalmente en desacuerdo con la afirmación presentada y 5 totalmente de acuerdo. (Véase Anexo G)

Este cuestionario se divide en dos partes. La primera parte mide la usabilidad general de la aplicación y se ha empleado el System usability scale (SUS) [43], el cual se obtiene empleando un cuestionario formado por 10 preguntas que están emparejadas entre sí. Este sistema de par de preguntas consiste en dos preguntas similares que tratan de lo mismo pero con significados opuestos, es decir, mientras que la primera afirmación indica una cosa como pueda ser “Encuentro que la aplicación es muy difícil de usar”, su pregunta emparejada sería “Creo que la aplicación fue fácil de usar”. De esta manera, las afirmaciones positivas suman el valor puntuado por el usuario menos uno y las afirmaciones negativas suman cinco menos el valor puntuado. Este cuestionario obtiene su puntuación de la formula 5.1 siendo P_1 la puntuación de las afirmaciones positivas y P_2 la puntuación de las afirmaciones negativas.

$$Puntuacion = 2,5 \cdot \sum ((P_1 - 1) + (5 - P_2)) \quad (5.1)$$

La segunda parte del cuestionario mide la satisfacción de elementos concretos de la aplicación. Para este apartado también se ha hecho uso del sistema de preguntas pareadas y se analizarán por separado. Estas preguntas nos permiten conocer de una manera más exacta la calidad y el nivel de

satisfacción de ciertos aspectos clave de la aplicación. En este apartado se ha decidido medir los siguientes aspectos:

- 1.– Tamaño del plugin con respecto a la pagina web.
- 2.– Tiempos de carga de la aplicación.
- 3.– Comprensión de la información mostrada en:
 - 3.1.– El código de colores y los gráficos presentados en la categoría de Análisis Emocional.
 - 3.2.– El modo de alto contraste.
 - 3.3.– Las entidades de la categoría de análisis de texto.
- 4.– La función de cada categoría por su título, icono y texto explicativo.
- 5.– La intuitividad de la posición del botón de atrás.
- 6.– Los colores usados en la aplicación.

Los resultados de esta encuesta se muestran en el Anexo H. Para la primera parte del cuestionario, siguiendo la fórmula propuesta para los cuestionarios SUS y promediando el resultado por el número de usuarios, obtenemos un resultado final de usabilidad 80,57 sobre 100. Este dato nos muestra que, a pesar de los puntos de mejora señalados por los usuarios, en términos de usabilidad, la aplicación cumple correctamente.

Para el apartado específico dividiremos el resultado por aspecto clave siguiendo la numeración presente en el listado anterior. El resultado de esta parte es sobre 10 y puede verse en la tabla 5.1. Para el cálculo de este resultado se ha seguido la misma fórmula presentada anteriormente pero individual para cada par de preguntas.

Aspecto	Resultado Ponderado
1	7,76
2	5,69
3.1	5,46
3.2	7
3.3	5,38
4	6,15
5	7,15
6	7,23

Tabla 5.1: Resultados de la parte específica del cuestionario de usabilidad

A continuación se listaran los principales fallos obtenidos tras la realización de las pruebas con usuarios, analizando el motivo por el cual se han podido dar y proponiendo una posible solución.

Empezando por los resultados favorables, podemos ver que en ciertos aspectos se consigue un valor alto de satisfacción. Estos aspectos son el tamaño del plugin con respecto a la página web, la claridad de la información con el modo de alto contraste activado, la posición del botón de retroceso y la paleta de colores usada en la aplicación.

Con respecto a estos aspectos podemos ver que aún existe margen de mejora. Por ejemplo, para hacer más clara la información en la categoría de Análisis Emocional con el modo de alto contraste activado, podrían usarse colores con texturas. Esto haría más evidente la diferencia entre cada color de la escala y ayudaría a diferenciarlos.

Siguiendo con los fallos de la aplicación, como podemos ver en la tabla 5.1, existen ciertos puntos que están claramente por debajo de la media. La aplicación es tosca en ciertos aspectos y, aun con potencial, requiere de un nuevo proceso de refinamiento para poder sacarse al mercado. Los fallos más evidentes son los errores de código. Existen varios de estos errores, presentes a lo largo de toda la aplicación, que nublan la experiencia de uso.

Tras estos errores podemos ver otros cuantos que son de diseño. Comenzando por los gráficos de la categoría de Análisis Emocional, podemos ver que necesita ser replanteada su presentación. El gráfico de polaridad es confuso, su título poco ilustrativo y el diferente uso o significado que se le da a la paleta de colores para cada gráfico confunde bastante a los usuarios. Además, debería incluirse una pequeña descripción de cada gráfico, aunque fuese de manera opcional, con algún bloque de texto escondido para que el usuario pueda leer la explicación detallada del gráfico en vez de tener que intuirlo.

Continuando por otro aspecto de diseño, la elección de la palabra *Entidades* por su traducción directa del inglés *enentity* no ha sido acertada. El significado de esta palabra es poco claro. Además, debe seguir una misma nomenclatura pues en la categoría de Datos Básicos existe el tipo de entidad *Otros* pero en la categoría de Análisis del texto, a ese tipo se le denomina *Misceláneo*. Otro gran problema que ha sufrido este aspecto es un efecto colateral de los fallos de precisión de las librerías usadas. Esto se debe principalmente a que se han usado librerías de carácter genérico y/o con pobre traducción al español. Para mejorar este aspecto, a parte de referirnos a él por otro nombre, deberíamos configurar la librería de manera específica para nuestro caso de uso e intentar conseguir reducir la tasa de error.

El siguiente error en el diseño es el título escogido para algunas categorías. En la categoría de Análisis Emocional, a algunos usuarios no le ha quedado claro por qué polaridad y subjetividad se encontraban en esa categoría. Con respecto a la categoría de Análisis del Texto, la mayoría de los encuestados no supieron explicar de qué se trataba la categoría o por qué se llamaba así. Claramente esta categoría debe ser repensada pues carece de una funcionalidad clara y de peso como para ser considerada categoría. Si no se le añade funcionalidad, podría pasar a ser un modo opcional de visualización de las entidades en la categoría de Datos Básicos. Por último, el propósito de la categoría

de Noticia Accesible no era claro y los usuarios no supieron ver su utilidad.

Por último, veremos dos fallos con respecto a la facilidad de uso y navegación que tiene la aplicación. Estos fallos son, la forma de mostrar las categorías en la pantalla de selección de categorías y los tiempos de carga de la aplicación. El problema principal de la aplicación con respecto a los tiempos de carga es la falta de indicación visual de que la aplicación esta cargando. Eso llevaba a pensar al usuario que la aplicación no funciona. Con respecto a la pantalla de selección de categorías, la idea principal fue crear un sistema de visualización que fuese escalable a muchas categorías. Por ello, se pensó en apilar las categorías en columnas y que fuesen visibles a través de una barra de scroll. El problema de esta barra [44] es que está escondida por defecto y, a no ser que se pase el cursor por encima de esta, y sin pasarlo muy rápido, no existe indicación alguna para el usuario de que existen más categorías que no está viendo actualmente. Por suerte, este fallo es tiene fácil solución, pues el componente usado para estilizar esta barra es bastante personalizable.

Como conclusión de esta sección, podemos ver que la aplicación a tenido una aceptación moderadamente positiva. Los usuarios ven el potencial, la información presentada es útil, pero tiene algunos fallos que necesitan ser pulidos para poder llegar a un público general.

CONCLUSIONES Y TRABAJO FUTURO

Para finalizar, en este capítulo se comentarán las conclusiones finales del proyecto y posibles ideas para el futuro de Transparency Plugin.

6.1. Conclusiones

Concluyendo este proyecto podemos decir que se ha conseguido un gran avance. Si nos remontamos la Introducción y el concepto de transparencia periodística podemos ver que Transparency Plugin cumple con la idea y la desarrolla un poco más allá. Esta herramienta puede ser útil tanto para el usuario medio como para un investigador pues aporta muchos datos indirectos que pueden ser interesantes. Con las pruebas con los usuarios me he dado cuenta de que la aplicación necesita trabajo. Esta claro que si quiere cumplir el objetivo de ser una herramienta amigable y satisfactoria, se deben pulir algunos bordes afilados pero el concepto ya esta en marcha y funcionando.

A modo de conclusión personal puedo decir que este Trabajo Final de Grado ha supuesto un reto. Nunca había realizado un proyecto de esta envergadura y menos pasando por todas las fases del proyecto desde la fase de análisis y estudio de mercado hasta la de pruebas con usuarios. Algo en lo que no se profundiza demasiado en la carrera son los test de usabilidad y satisfacción. Gracias a este proyecto he podido conocer un poco más sobre este apartado que, aunque a veces se deje de lado, es una de las mayores fuentes de información sobre fallos y posibles mejoras para la aplicación.

Sin duda he cumplido uno de mis objetivos personales que era aprender una tecnología nueva. Angular ha resultado ser un framework mucho más potente de lo que pensaba en un principio y, aunque no he podido usar muchas de sus herramientas que lo hacen potente, he podido conocerlas y trastear con ellas. Sin duda, los conocimientos que he aprendido con este proyecto los usaré durante mi desarrollo profesional como Ingeniero Informático.

6.2. Trabajo Futuro

Durante el desarrollo de este proyecto, han pasado por mi cabeza miles de ideas nuevas que se podrían implementar en la aplicación en un futuro. Dejando de lado el hecho de que lo primero que hay que hacer es arreglar todos los fallos y conseguir un producto perfecto con la funcionalidad que tiene actualmente, existe un margen de mejora y ampliación muy alto.

De las primeras cosas que se me ocurren, es llevar esta herramienta a distintos navegadores. Actualmente se optó por desarrollarla para Chrome pero el cambio entre navegadores no es muy complicado y el plugin es escalable a otros navegadores. Otra inclusión que conllevaría apenas cambio en el código pero un gran impacto en los usuarios es permitir otros idiomas, tanto para la aplicación como para las noticias a analizar.

Por su puesto, las librerías y herramientas que usa actualmente la aplicación son gratuitas y la mayoría de carácter genérico. En varias de ellas como SpaCy, tenemos la posibilidad de incluir nuestras propias reglas o redes neuronales para el análisis, lo que supondría una reducción enorme de la tasa de error que tiene actualmente. De la misma manera que SpaCy otras herramientas como el análisis de las emociones tienen un gran margen de mejora si se esta dispuesto a pagar un poco o dedicar un desarrollo a pulir la herramienta.

BIBLIOGRAFÍA

- [1] B. C. Martisi, “La transparencia, herramienta básica para el periodismo de calidad,” Online en <https://www.compromisoempresarial.com/transparencia/2017/12/la-transparencia-herramienta-basica-para-el-periodismo-de-calidad/>, 2017 (Accedido el 04-07-2020).
- [2] “Así funciona la calculadora de transparencia editorial de ‘Público’,” Online en <https://blogs.publico.es/publico/2018/10/04/asi-funciona-la-calculadora-de-transparencia-editorial-de-publico/>, 2018 (Accedido el 04-07-2020).
- [3] “Text Analytics API, Documentación Oficial,” Online en <https://azure.microsoft.com/es-es/services/cognitive-services/text-analytics/>, 2020 (Accedido el 07-04-2020).
- [4] “Social Media Macroscope, Documentación Oficial,” Online en <https://socialmediamacroscope.org/>, 2020 (Accedido el 04-07-2020).
- [5] “Angular, Documentación Oficial,” Online en <https://angular.io/>, 2020 (Accedido el 04-07-2020).
- [6] B. Tahir, “Building A ‘Serverless’ Chrome Extension,” Online en <https://towardsdatascience.com/building-a-serverless-chrome-extension-f684740e1ffc>, 2019 (Accedido el 04-07-2020).
- [7] “Usage share of web browsers,” Online en https://en.wikipedia.org/wiki/Usage_share_of_web_browsers, 2020 (Accedido el 04-07-2020).
- [8] “Learn Angular - Angular 8 path courses,” Online en <https://www.pluralsight.com/paths/angular>, 2020 (Accedido el 04-07-2020).
- [9] B. Chavan, “Angular + Python + Flask — Full stack demo,” Online en <https://medium.com/@balramchavan/angular-python-flask-full-stack-demo-27192b8de1a3>, 2018 (Accedido el 04-07-2020).
- [10] A. Caamal, “Flask / Angular 8 / MySQL #1 (Installing dependencies and creating the DB),” Online en https://www.youtube.com/watch?v=ZZ60XeA922c&list=PLolX_BtuGc9SCj7sTXSABVn-B1vb5EPQy, 2019 (Accedido el 04-07-2020).
- [11] B. Krebs, “Using Python, Flask, and Angular to Build Modern Web Apps - Part 1,” Online en <https://auth0.com/blog/using-python-flask-and-angular-to-build-modern-apps-part-1/>, 2018 (Accedido el 04-07-2020).
- [12] “RxJS, Documentación Oficial,” Online en <https://rxjs-dev.firebaseapp.com/>, 2020 (Accedido el 04-07-2020).
- [13] “Angular Material UI component library,” Online en <https://material.angular.io/>, 2020 (Accedido el 04-07-2020).
- [14] “Mat-Icon List,” Online en <https://www.angularjswiki.com/angular/angular-material-icons-list-mat-icon-list/>, 2020 (Accedido el 04-07-2020).

- [15] "Jasmine, Documentación Oficial," Online en <https://jasmine.github.io/>, 2020 (Accedido el 04-07-2020).
- [16] "Material Design, Documentación Oficial," Online en <https://material.io/design/>, 2020 (Accedido el 04-07-2020).
- [17] "Bootstrap, Documentación Oficial," Online en <https://getbootstrap.com/docs/4.1/getting-started/introduction/>, 2020 (Accedido el 04-07-2020).
- [18] "Highcharts, Documentación Oficial," Online en <https://www.highcharts.com/>, 2020 (Accedido el 04-07-2020).
- [19] Asymmetrik, "Asymmetrik/ngx-leaflet: Core Leaflet package for Angular.io," Online en <https://github.com/Asymmetrik/ngx-leaflet>, 2019 (Accedido el 04-07-2020).
- [20] "Karma, Documentación Oficial," Online en <https://karma-runner.github.io/latest/index.html>, 2020 (Accedido el 04-07-2020).
- [21] Y. M. Jiménez, "Haciendo tests en Angular, con Jasmine y Karma," Online en <https://medium.com/@yonem9/haciendo-tests-en-angular-con-jasmine-y-karma-5b378368714f>, 2018 (Accedido el 04-07-2020).
- [22] "Flask, Documentación Oficial," Online en <https://flask.palletsprojects.com/en/1.1.x/>, 2020 (Accedido el 04-07-2020).
- [23] "Flask-RESTful, Documentación Oficial," Online en <https://flask-restful.readthedocs.io/en/latest/>, 2020 (Accedido el 04-07-2020).
- [24] "Flask-Caching, Documentación Oficial," Online en <https://flask-caching.readthedocs.io/en/latest/>, 2020 (Accedido el 04-07-2020).
- [25] "Newspaper3k, Documentación Oficial," Online en <https://newspaper.readthedocs.io/en/latest/>, 2020 (Accedido el 04-07-2020).
- [26] "SpaCy, Documentación Oficial," Online en <https://spacy.io/>, 2020 (Accedido el 04-07-2020).
- [27] University of Southampton, "geoparsepy · PyPI," Online en <https://pypi.org/project/geoparsepy/>, 2019 (Accedido el 04-07-2020).
- [28] "TextBlob, Documentación Oficial," Online en <https://textblob.readthedocs.io/en/dev/>, 2020 (Accedido el 04-07-2020).
- [29] National Research Council of Canada, "NRC Emotion Lexicon," Online en <https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>, 2016 (Accedido el 04-07-2020).
- [30] "Visual Studio Code, Documentación Oficial," Online en <https://code.visualstudio.com/>, 2020 (Accedido el 04-07-2020).
- [31] "GitHub," Online en <https://github.com/>, 2020 (Accedido el 04-07-2020).
- [32] "NPM, Documentación Oficial," Online en <https://www.npmjs.com/>, 2020 (Accedido el 04-07-2020).
- [33] "Trello," Online en <https://trello.com/es>, 2020 (Accedido el 04-07-2020).
- [34] "PlantUML, Documentación Oficial," Online en <https://plantuml.com/es/>, 2020 (Accedido el 04-07-2020).

- [35] “Desarrollo iterativo e incremental,” Online en <https://proyectosagiles.org/desarrollo-iterativo-incremental/>, 2020 (Accedido el 04-07-2020).
- [36] N. Blanco, “¿Qué patron usa Angular? MVC o MVVM,” Online en <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/>, 2018 (Accedido el 04-07-2020).
- [37] L. Marx, “Is Angular 2+ MVVM?,” Online en <https://malcoded.com/posts/angular-2-components-and-mvvm/>, 2016 (Accedido el 04-07-2020).
- [38] “Redux, Documentación Oficial,” Online en <https://es.redux.js.org/>, 2020 (Accedido el 04-07-2020).
- [39] R. Cravit, “How to Use Color Blind Friendly Palettes to Make Your Charts Accessible,” Online en <https://venngage.com/blog/color-blind-friendly-palette/>, 2019 (Accedido el 04-07-2020).
- [40] Alir3z4, “Alir3z4/stop-words: List of common stop words in various languages,” Online en <https://github.com/Alir3z4/stop-words>, 2020 (Accedido el 04-07-2020).
- [41] “World Wide Web Consortium (W3C), Documentación Oficial,” Online en <https://www.w3.org/>, 2020 (Accedido el 04-07-2020).
- [42] “Web Content Accessibility Guidelines (WCAG) 2.0, Documentación Oficial,” Online en <https://www.w3.org/TR/WCAG20/>, 2020 (Accedido el 04-07-2020).
- [43] J. Sauro, “Measuring Usability with the System Usability Scale (SUS),” Online en <https://measuringu.com/sus/>, 2011 (Accedido el 04-07-2020).
- [44] Grsnto, “Grsnto/simplebar: Custom scrollbars vanilla javascript library with native scroll, done simple, lightweight, easy to use and cross-browser,” Online en <https://github.com/Grsnto/simplebar>, 2020 (Accedido el 04-07-2020).

APÉNDICES

DESCRIPCIÓN DE CASOS DE USO

ID:	CU-01
Caso de Uso:	Envío de Url a analizar
Actor:	Usuario
Descripción:	El usuario desea analizar una página web.
Precondición:	El usuario debe iniciar el plugin.
Flujo Principal:	<ol style="list-style-type: none">1.– El usuario introduce una Url en el formulario presentado.2.– El usuario envía el formulario rellenado.3.– El servidor recibe la Url y comprueba su validez.4.– El plug-in presenta una página con las categorías disponibles.
Flujo Alternativo:	<ol style="list-style-type: none">1.– El usuario analiza la página web en la que se encuentra haciendo click en un botón especial que analiza la página en la que el plug-in fue abierto.
Postcondición:	El usuario puede ver las categorías disponibles que contienen la información analizada y el plug-in guarda la Url enviada por el usuario.
Excepciones:	<ol style="list-style-type: none">1.– Si el usuario no introduce ninguna Url, o introduce una Url errónea, y envía el formulario mediante el botón principal, el plug-in mostrará un mensaje de error permitiendo volver a introducir una Url.

Tabla A.1: Caso de Uso 01

ID:	CU-02
Caso de Uso:	Seleccionar una categoría
Actor:	Usuario
Descripción:	El usuario desea seleccionar una categoría para ver su información.
Precondición:	El usuario debe haber enviado una Url al servidor y estar en el menú de selección de categorías.
Flujo Principal:	<ol style="list-style-type: none">1.– El usuario selecciona una categoría.2.– El servidor recibe la categoría seleccionada.3.– El servidor realiza el análisis de la Url almacenada obteniendo la información que debe contener esa categoría.4.– El plug-in muestra la información de la categoría al usuario.
Flujo Alternativo:	
Postcondición:	El plug-in muestra la información de la categoría seleccionada al usuario.
Excepciones:	<ol style="list-style-type: none">1.– El plug-in muestra un mensaje de error en caso de que el servidor dé error al analizar la Url.

Tabla A.2: Caso de Uso 02

ID:	CU-03
Caso de Uso:	Volver al menú de categorías
Actor:	Usuario
Descripción:	El usuario desea volver al menú de selección de categorías.
Precondición:	El usuario debe encontrarse en la página de alguna de las categorías.
Flujo Principal:	<ol style="list-style-type: none">1.– El usuario retrocede al menú de categorías.2.– El plug-in presenta las categorías disponibles.
Flujo Alternativo:	
Postcondición:	El plug-in muestra el menú de categorías disponibles al usuario.
Excepciones:	

Tabla A.3: Caso de Uso 03

ID:	CU-04
Caso de Uso:	Volver al formulario de introducción de Url a analizar
Actor:	Usuario
Descripción:	El usuario desea volver al formulario presentado en el CU-01.
Precondición:	El usuario debe haber rellenado el formulario presentado en el CU-01 y encontrarse en el menú de categorías.
Flujo Principal:	<ol style="list-style-type: none"> 1.– El usuario retrocede en la aplicación. 2.– El plug-in presenta el formulario para que el usuario lo rellene nuevamente.
Flujo Alternativo:	
Postcondición:	El plug-in muestra el formulario para analizar una Url.
Excepciones:	

Tabla A.4: Caso de Uso 04

ID:	CU-05
Caso de Uso:	Resaltar entidades en la categoría de Análisis del Texto
Actor:	Usuario
Descripción:	El usuario desea resaltar las entidades presentes en el texto dentro de la categoría de Análisis del Texto.
Precondición:	El usuario debe haber rellenado el formulario presentado en el CU-01 y encontrarse en la categoría de Análisis del Texto.
Flujo Principal:	<ol style="list-style-type: none"> 1.– El usuario hace clic en el botón de resaltar entidades. 2.– El plug-in resalta de manera gráfica las entidades presentes en el texto de la categoría.
Flujo Alternativo:	
Postcondición:	El usuario puede ver las entidades resaltadas de manera gráfica.
Excepciones:	

Tabla A.5: Caso de Uso 05

PRODUCT BACKLOG

B.1. Iteración 1

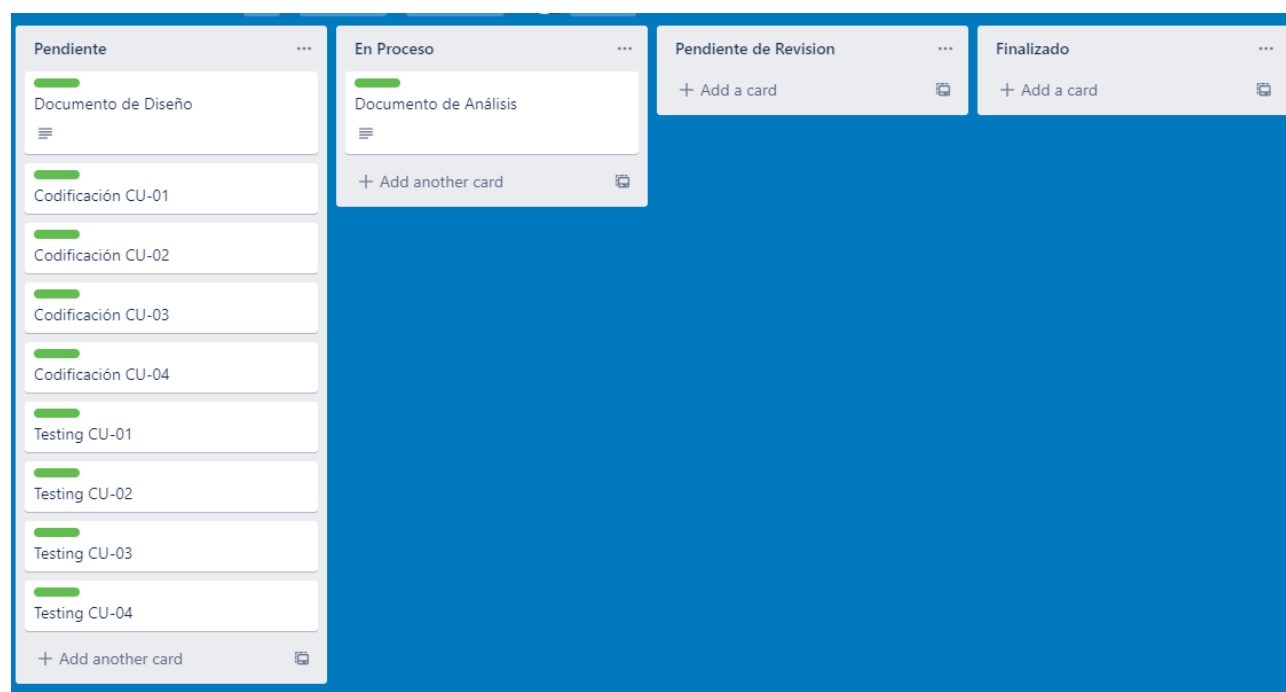


Figura B.1: Tablón de tareas de la iteración 1 - 1-10-19

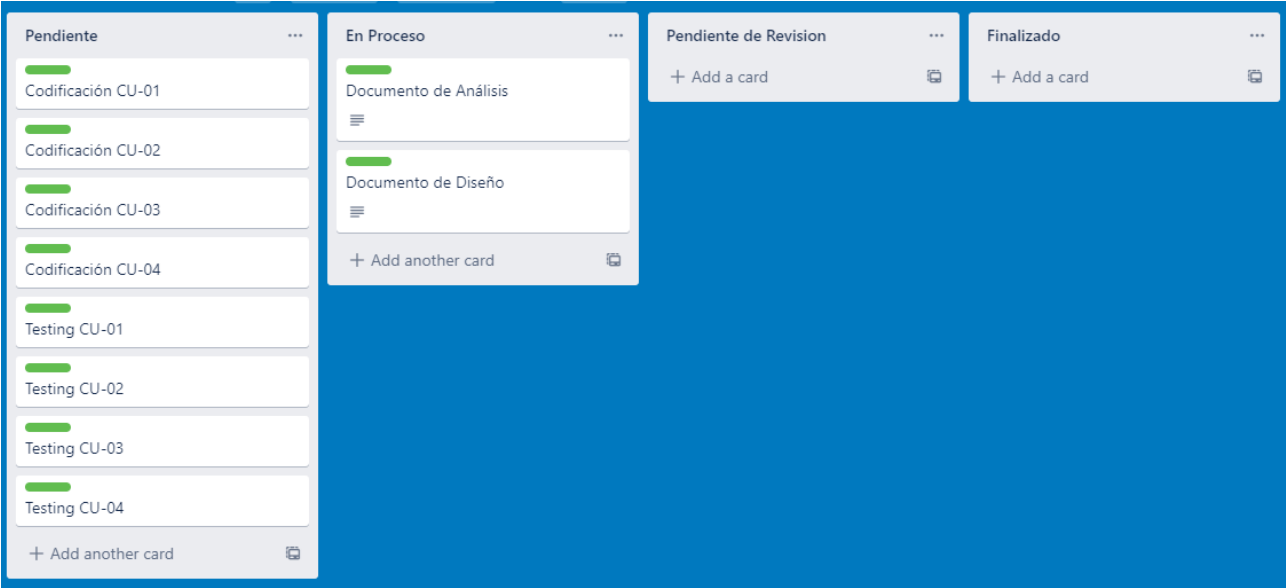


Figura B.2: Tablón de tareas de la iteración 1 - 5-10-19

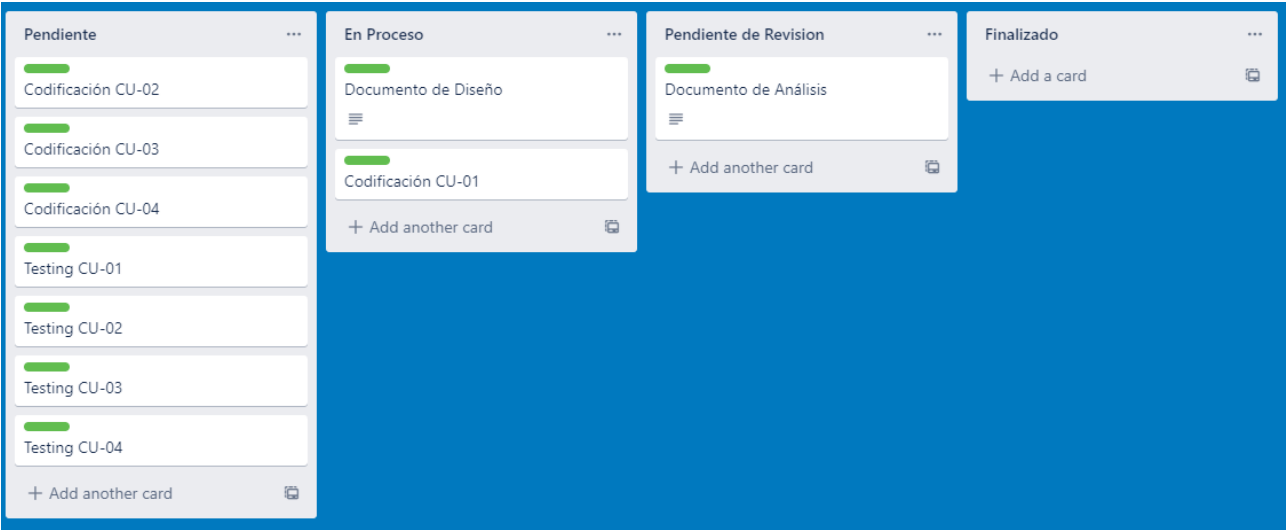


Figura B.3: Tablón de tareas de la iteración 1 - 6-10-19

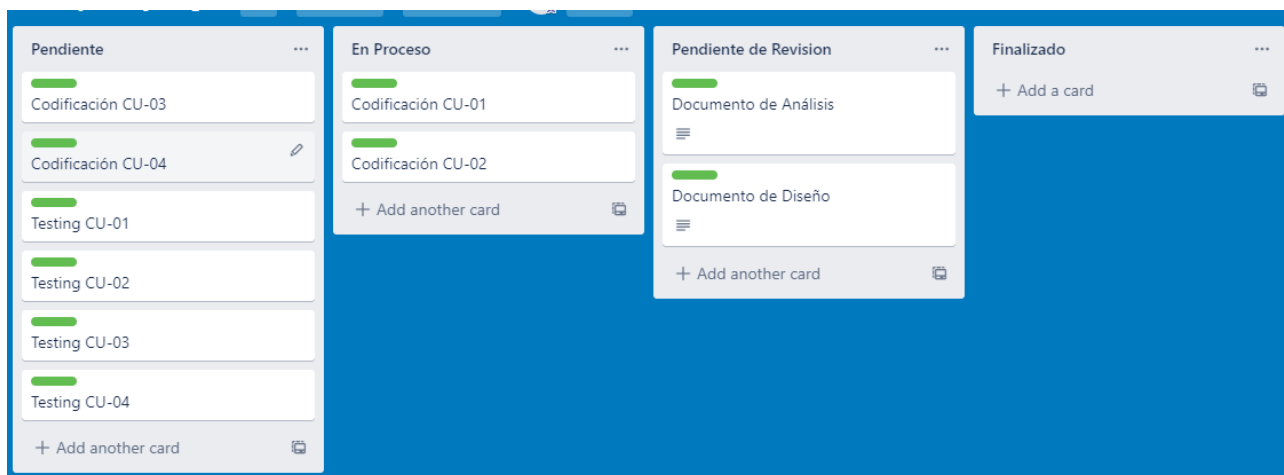


Figura B.4: Tablón de tareas de la iteración 1 - 10-10-19

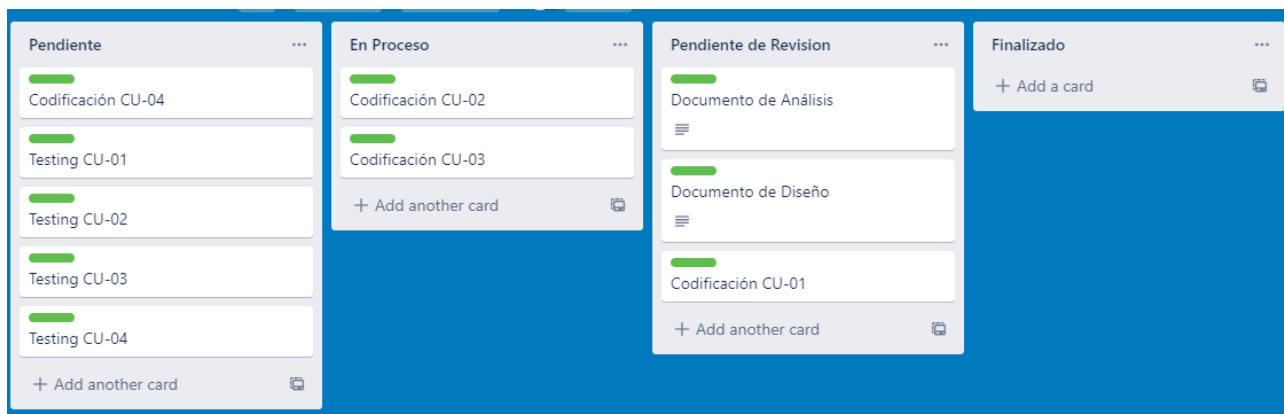


Figura B.5: Tablón de tareas de la iteración 1 - 2-11-19

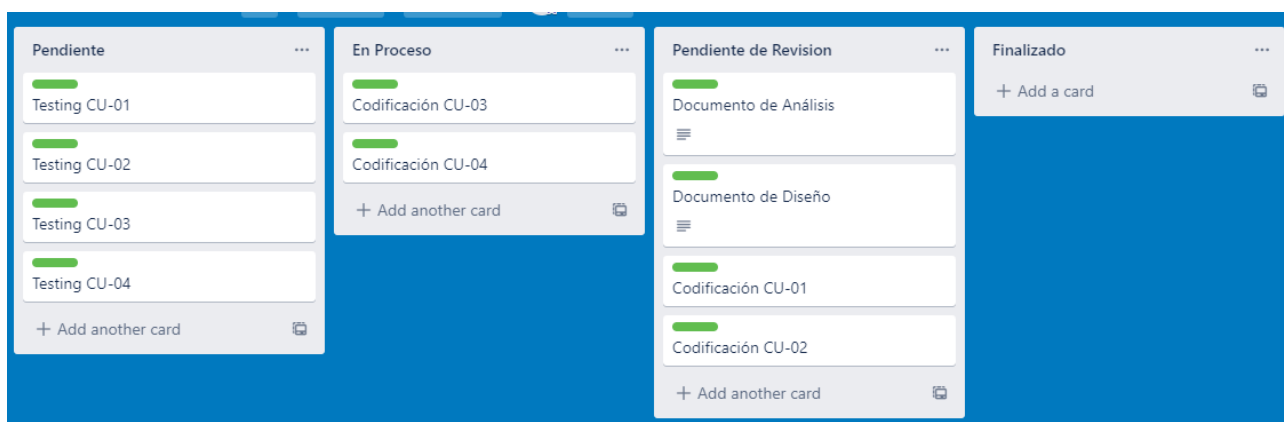


Figura B.6: Tablón de tareas de la iteración 1 - 26-12-19

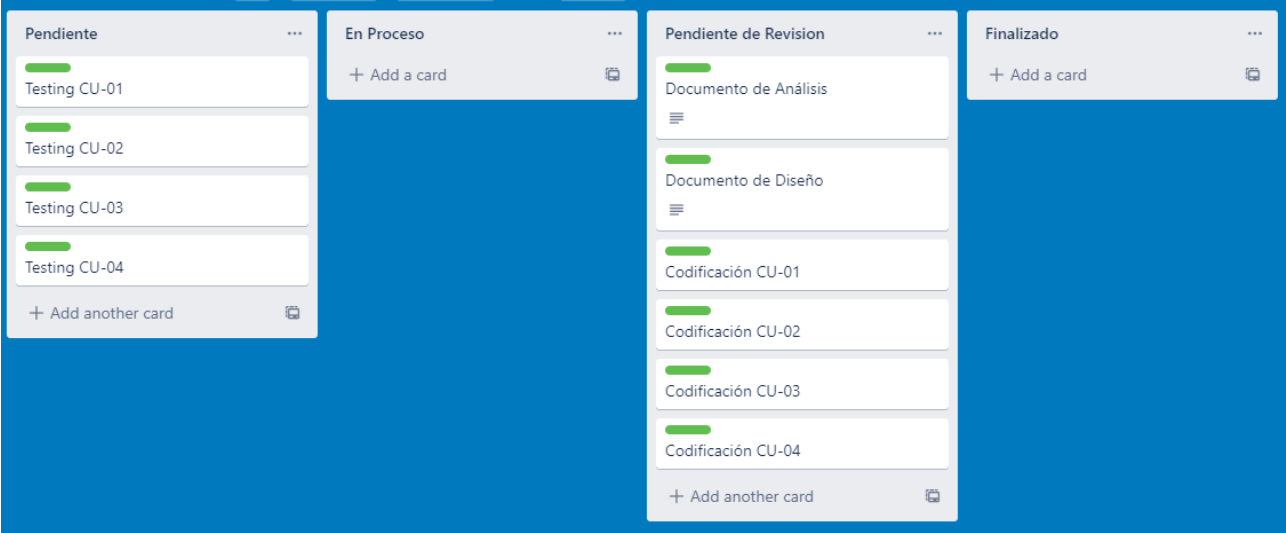


Figura B.7: Tablón de tareas de la iteración 1 - 30-12-19

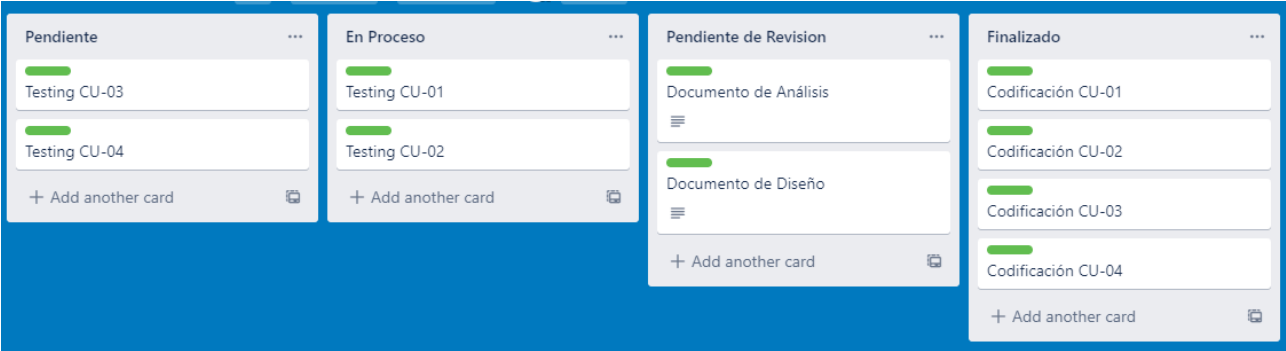


Figura B.8: Tablón de tareas de la iteración 1 - 5-1-20

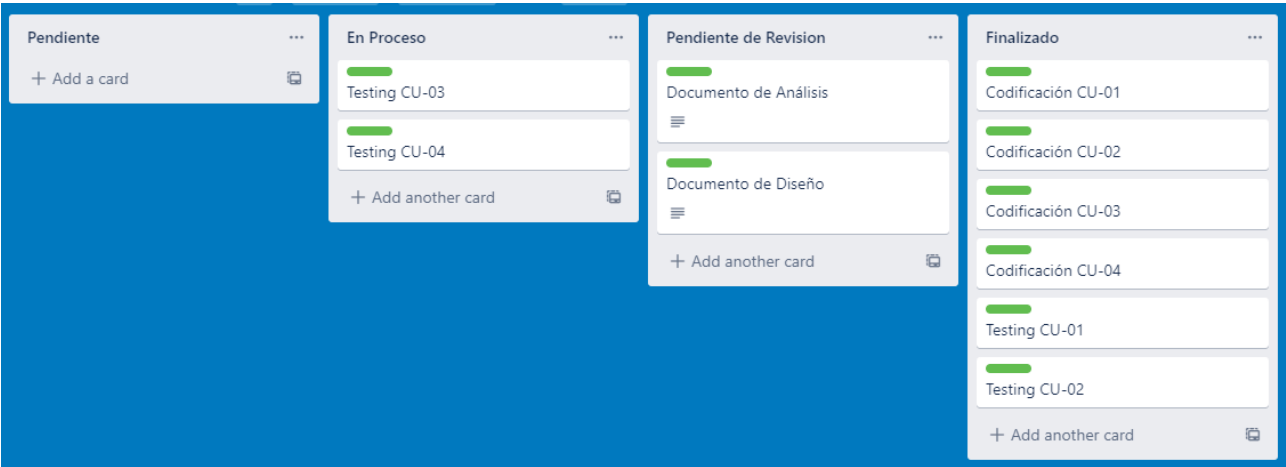


Figura B.9: Tablón de tareas de la iteración 1 - 7-1-20

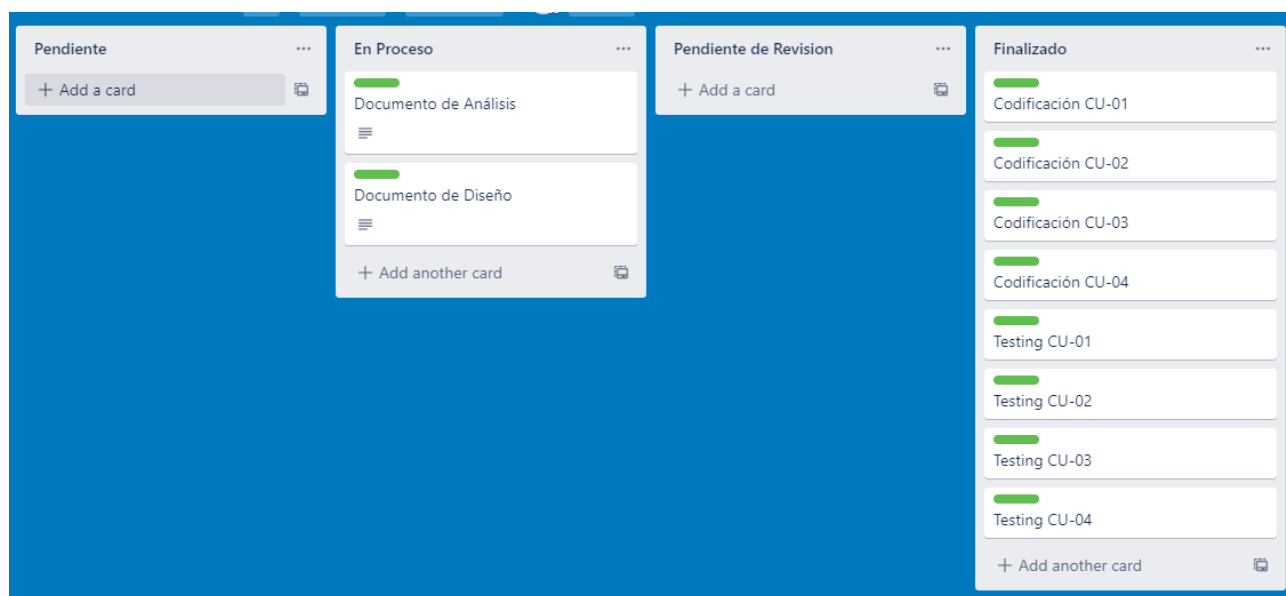


Figura B.10: Tablón de tareas de la iteración 1 - 14-1-20

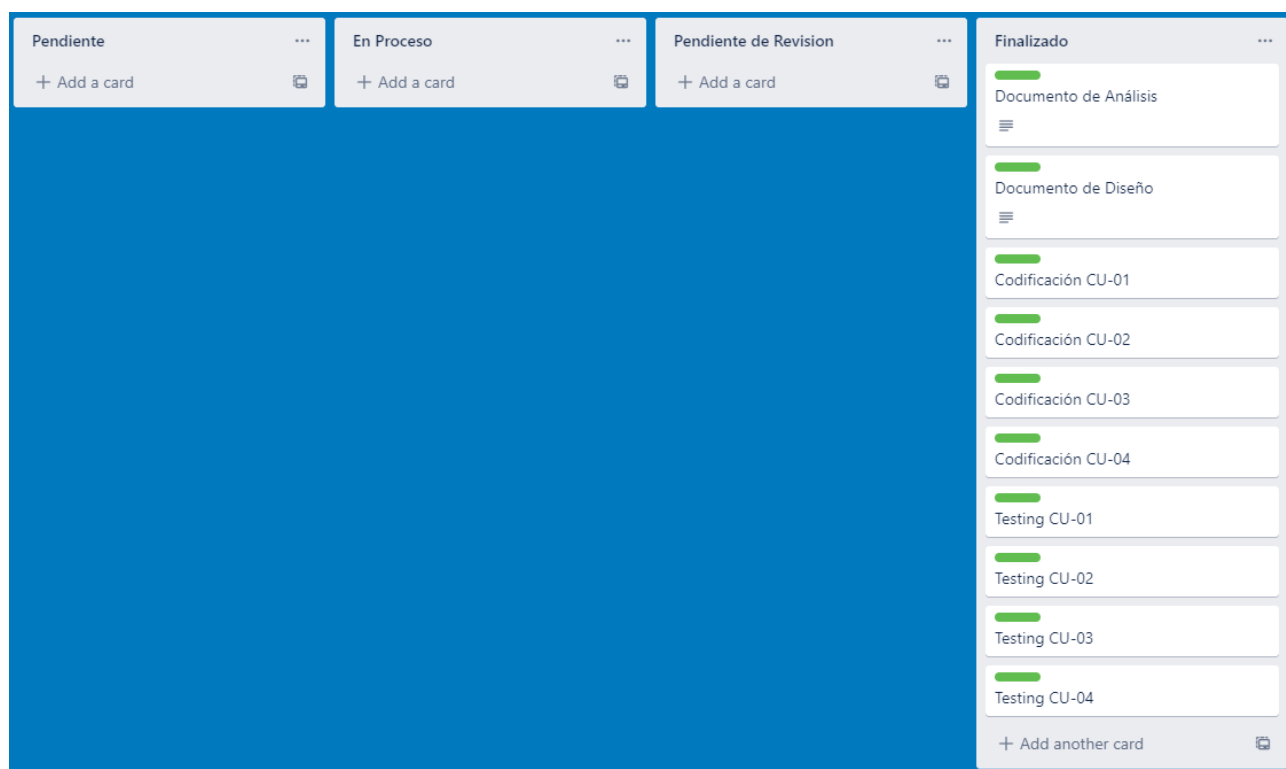


Figura B.11: Tablón de tareas de la iteración 1 - 24-1-20

B.2. Iteración 2

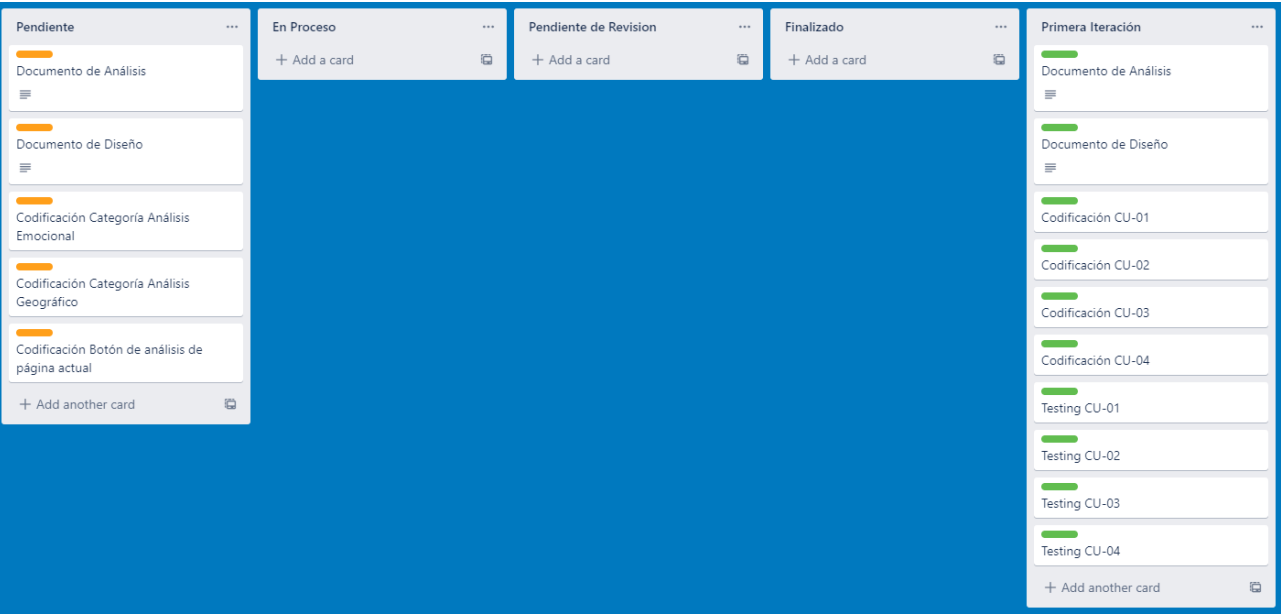


Figura B.12: Tablón de tareas de la iteración 2 - 25-1-20

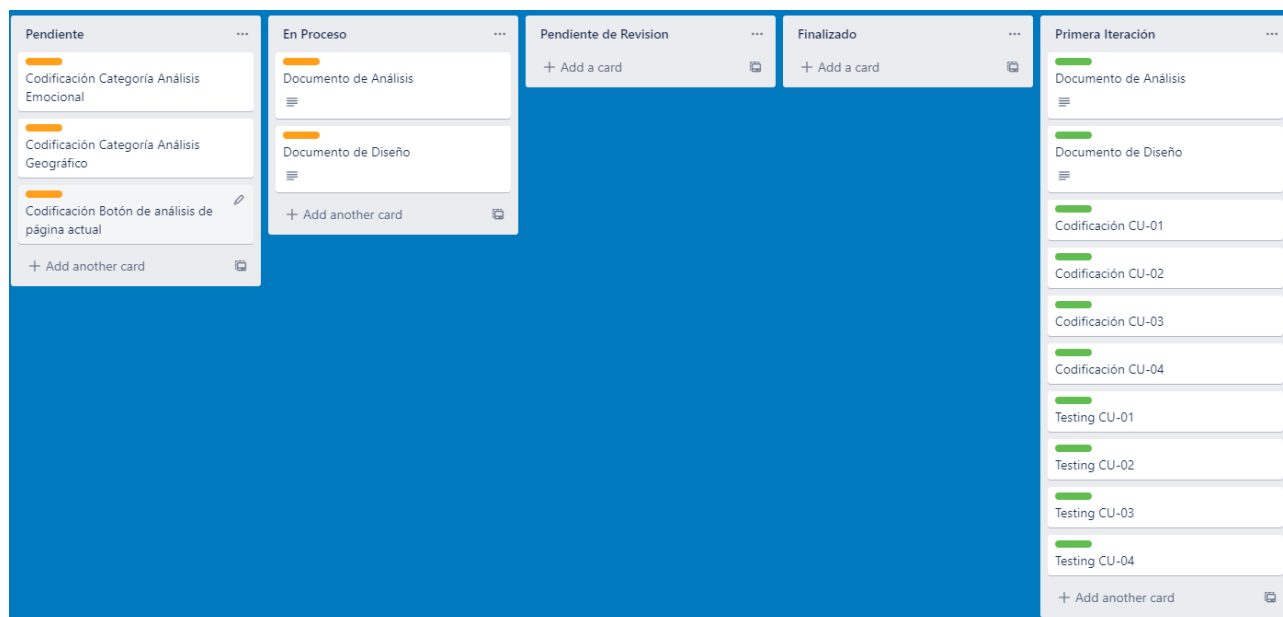


Figura B.13: Tablón de tareas de la iteración 2 - 26-1-20

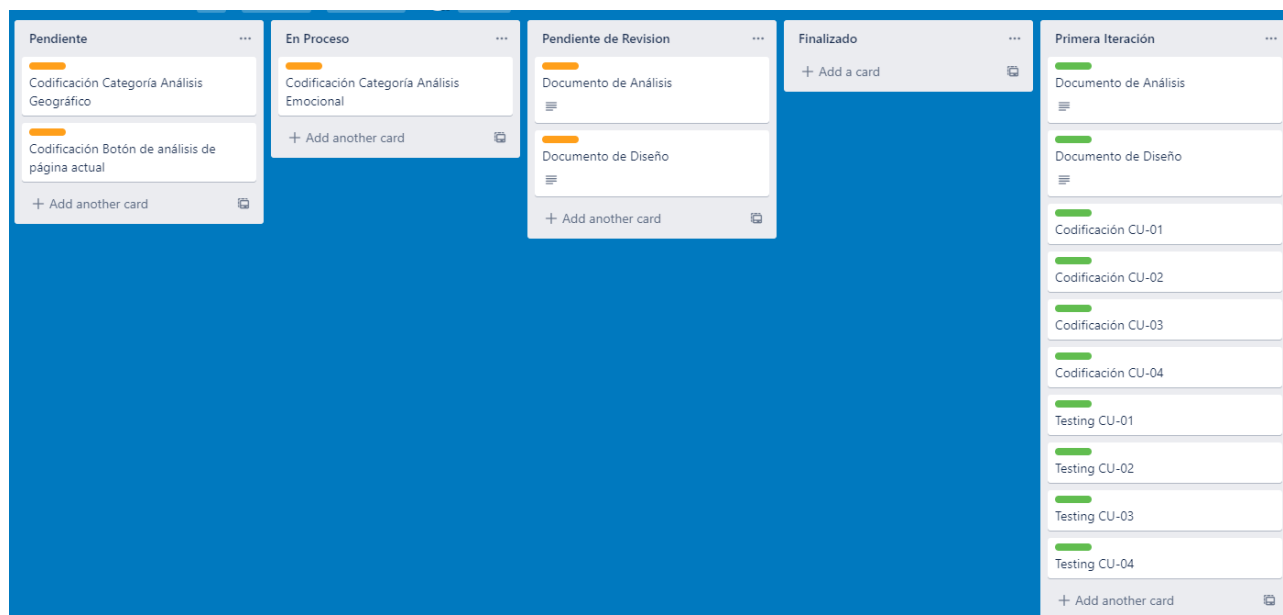


Figura B.14: Tablón de tareas de la iteración 2 - 29-1-20

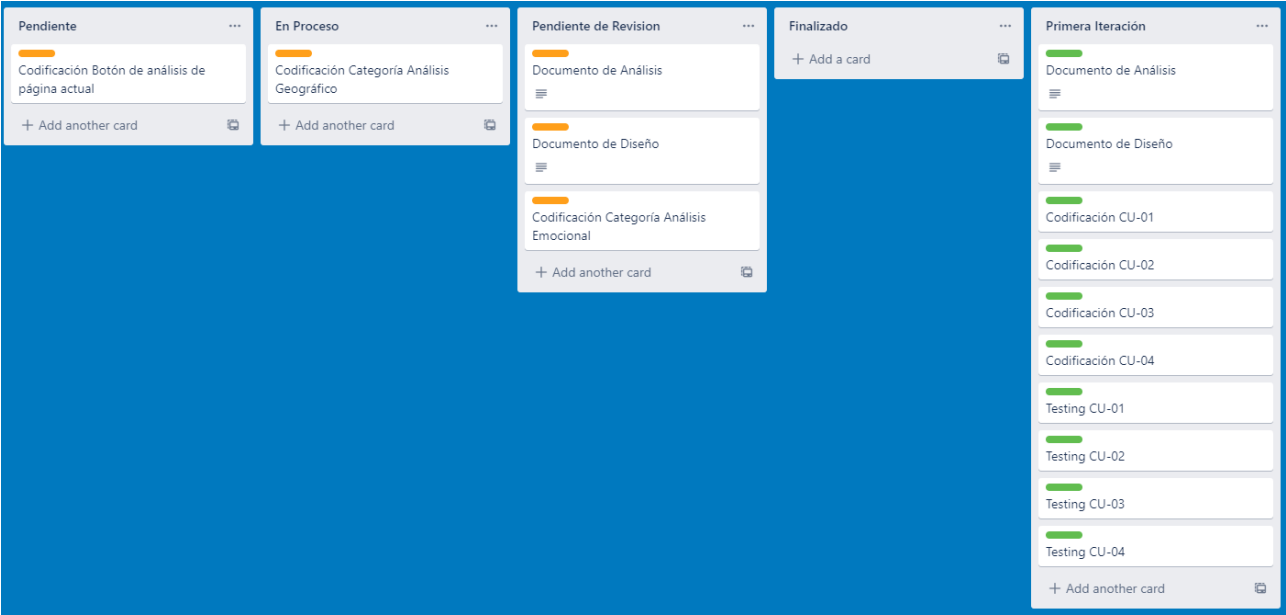


Figura B.15: Tablón de tareas de la iteración 2 - 6-2-20

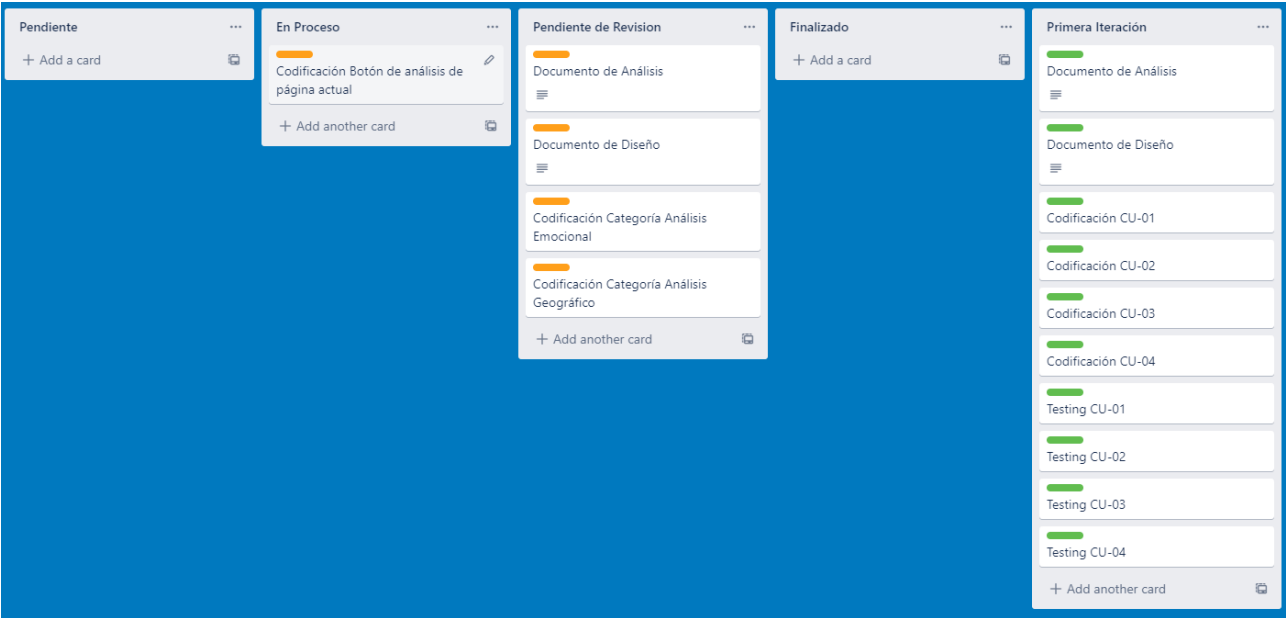


Figura B.16: Tablón de tareas de la iteración 2 - 12-2-20

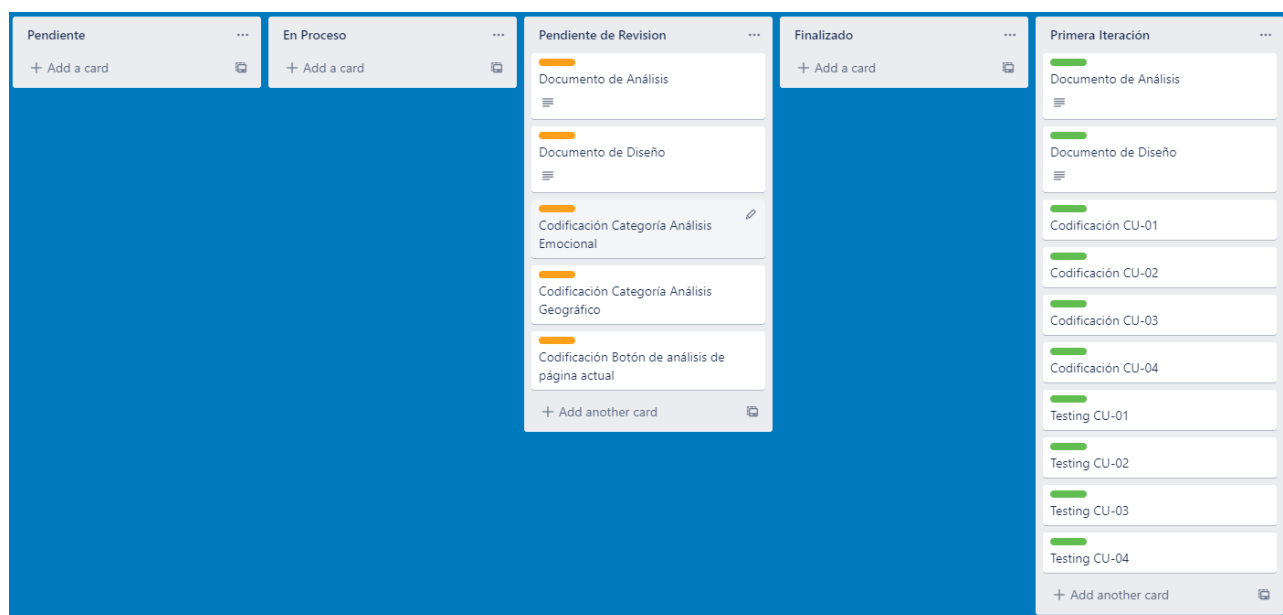


Figura B.17: Tablón de tareas de la iteración 2 - 13-2-20

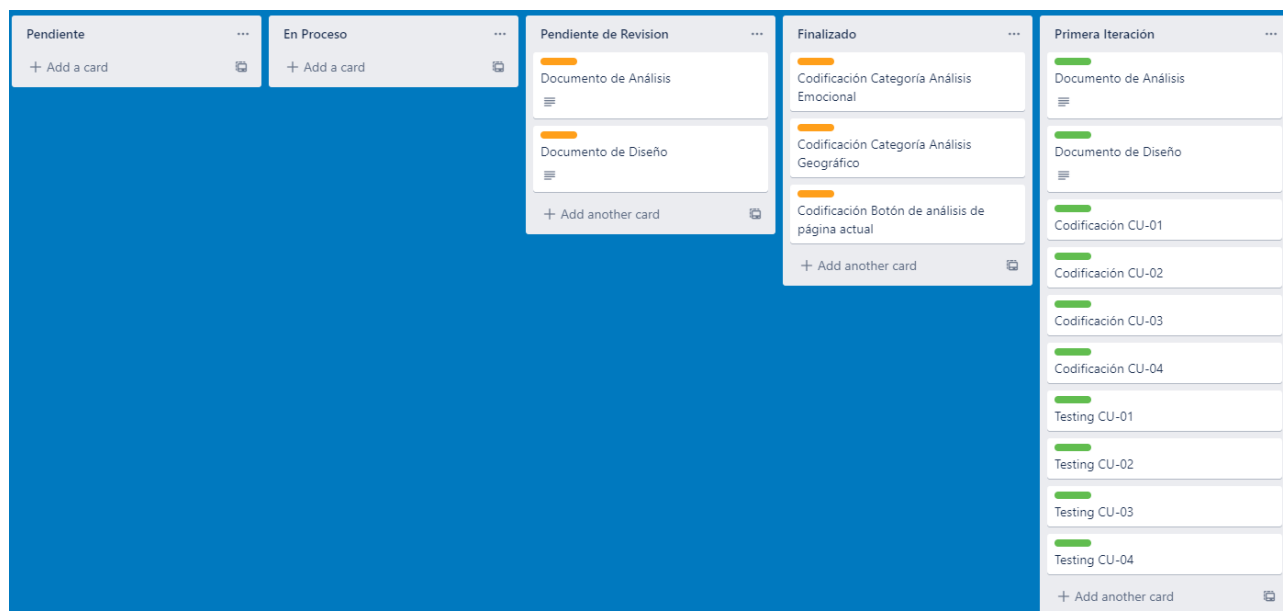


Figura B.18: Tablón de tareas de la iteración 2 - 14-2-20

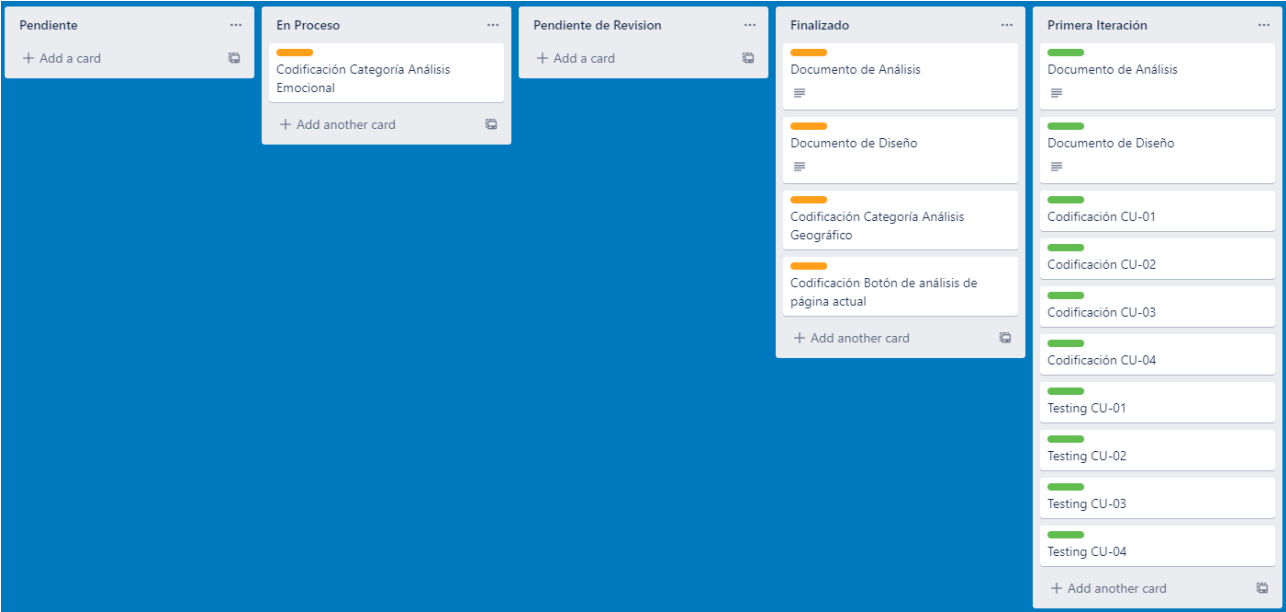


Figura B.19: Tablón de tareas de la iteración 2 - 16-2-20

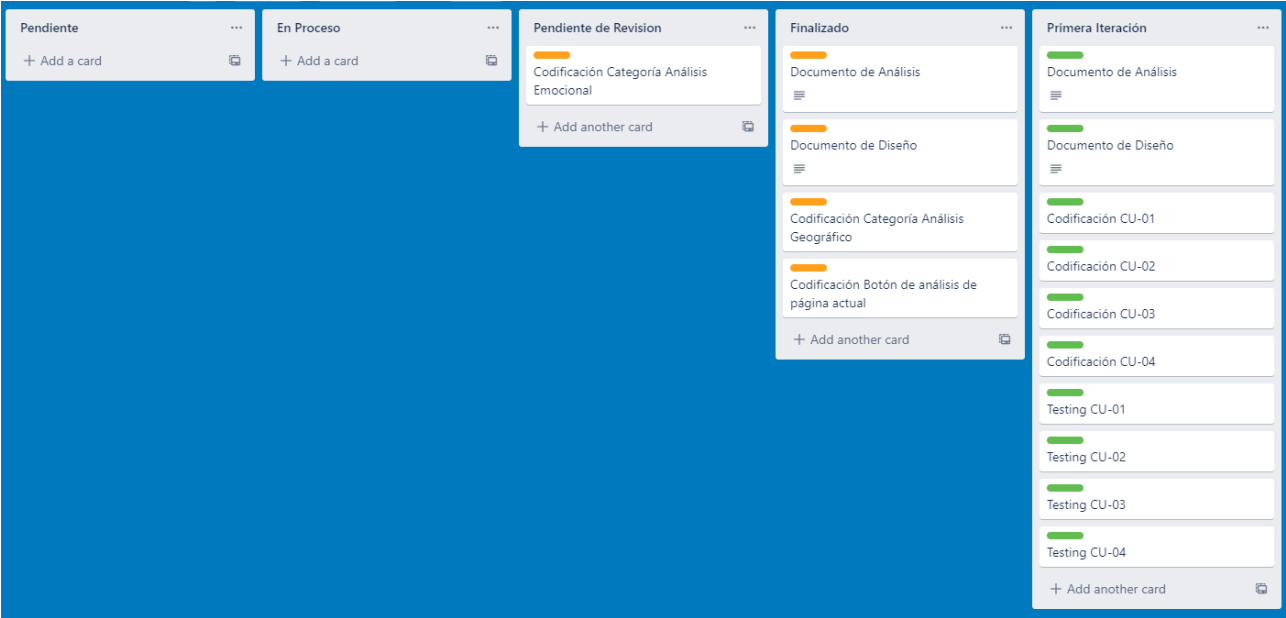


Figura B.20: Tablón de tareas de la iteración 2 - 26-2-20

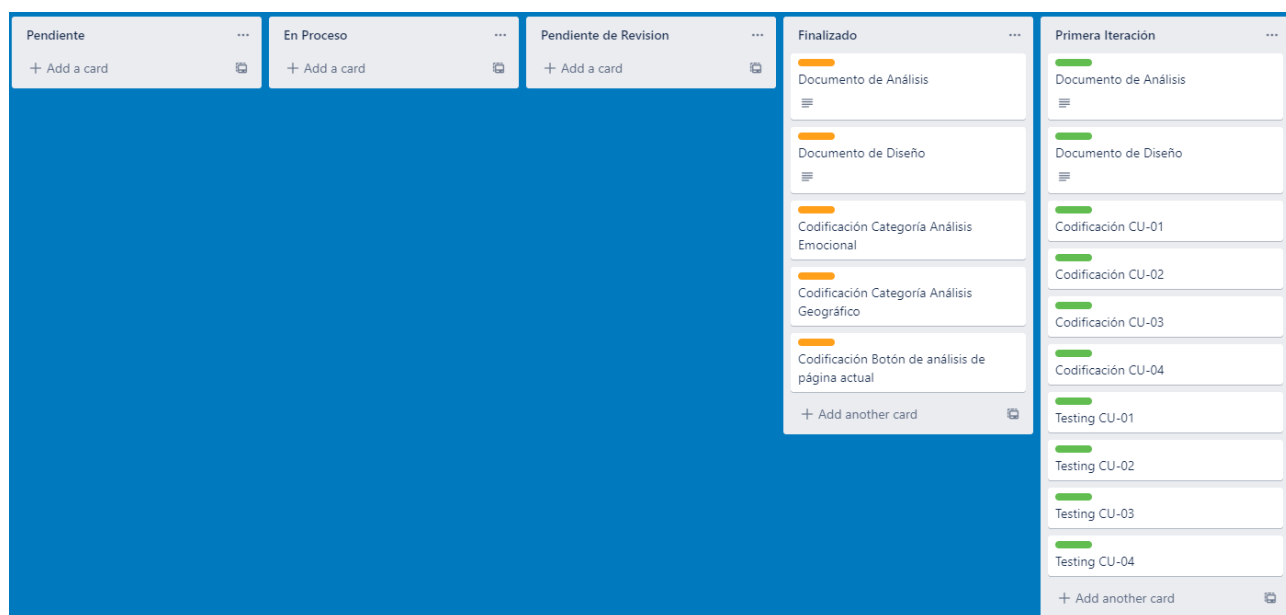


Figura B.21: Tablón de tareas de la iteración 2 - 30-2-20

B.3. Iteración 3

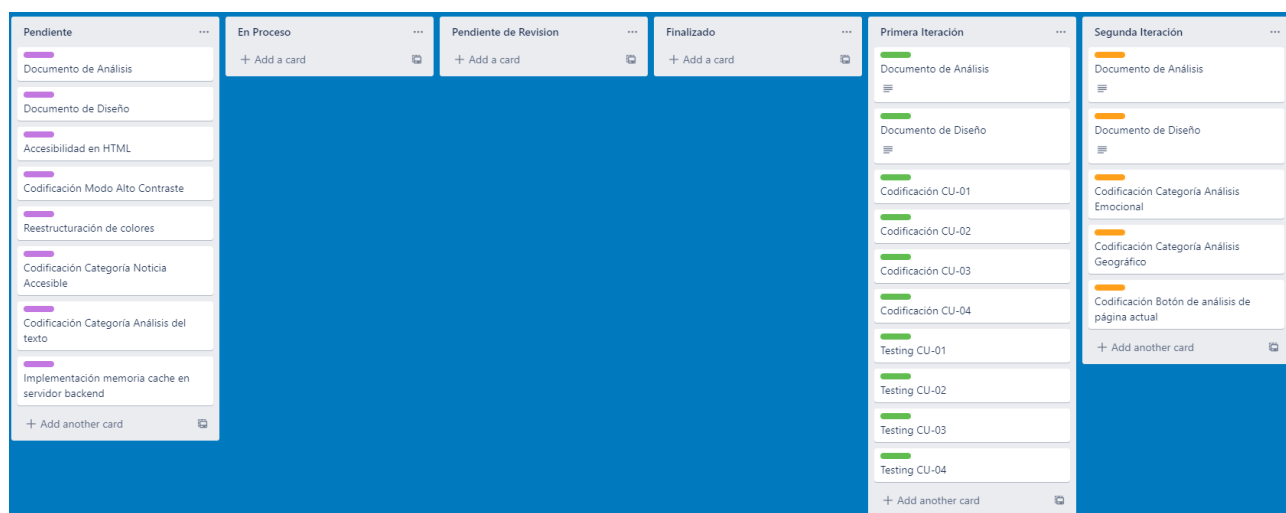


Figura B.22: Tablón de tareas de la iteración 3 - 27-2-20

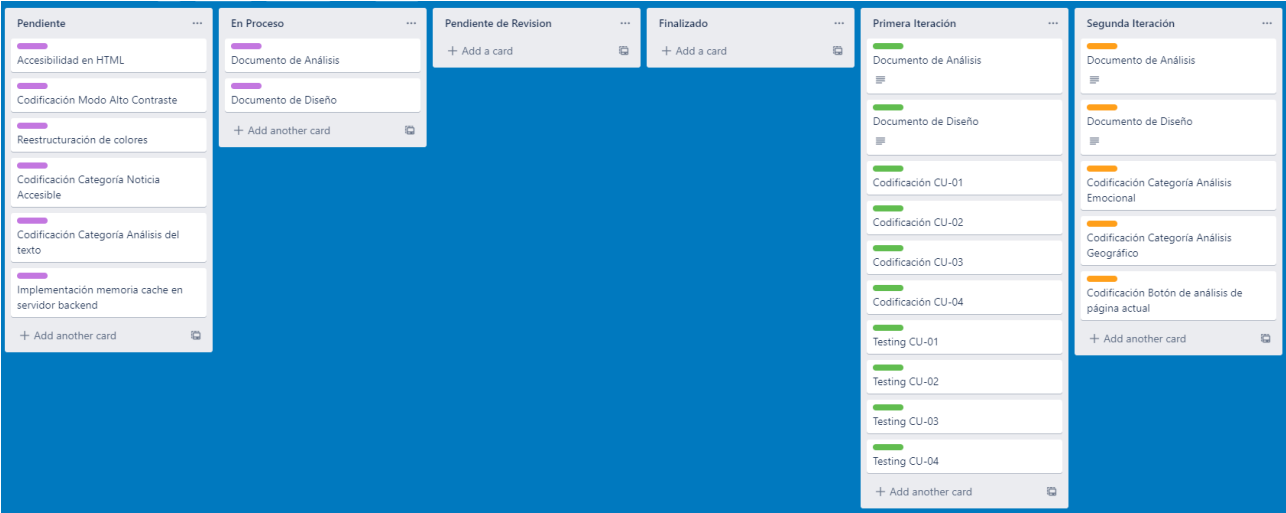


Figura B.23: Tablón de tareas de la iteración 3 - 28-2-20

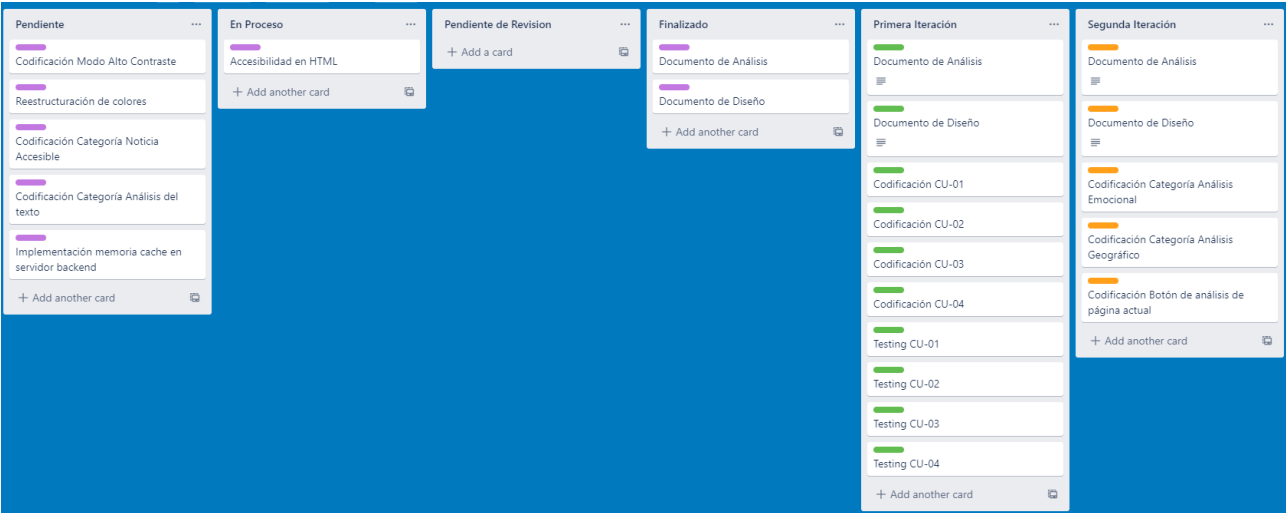


Figura B.24: Tablón de tareas de la iteración 3 - 1-3-20

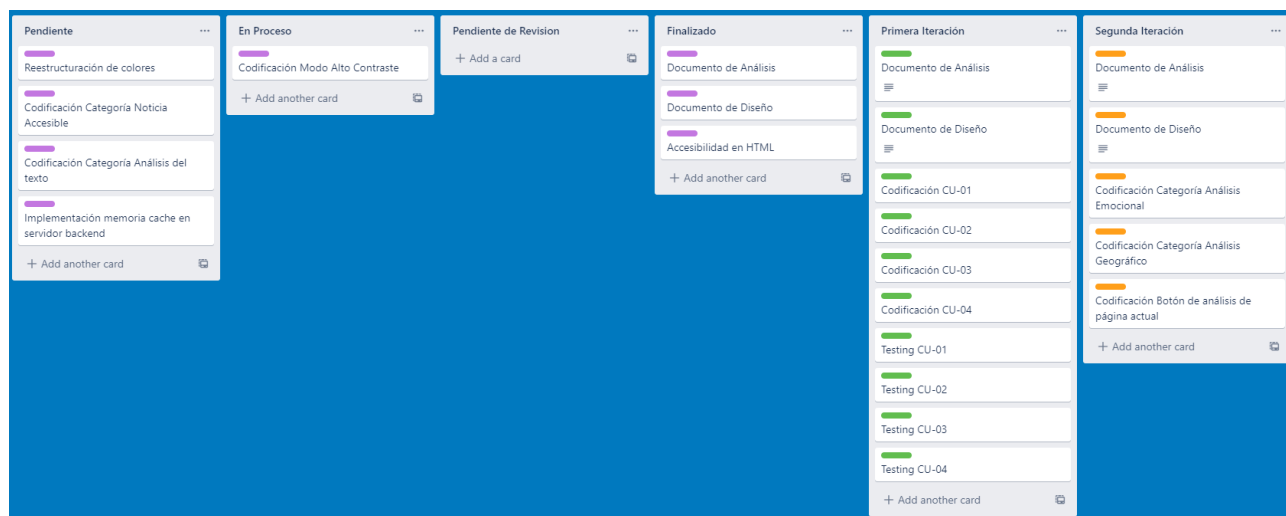


Figura B.25: Tablón de tareas de la iteración 3 - 2-3-20

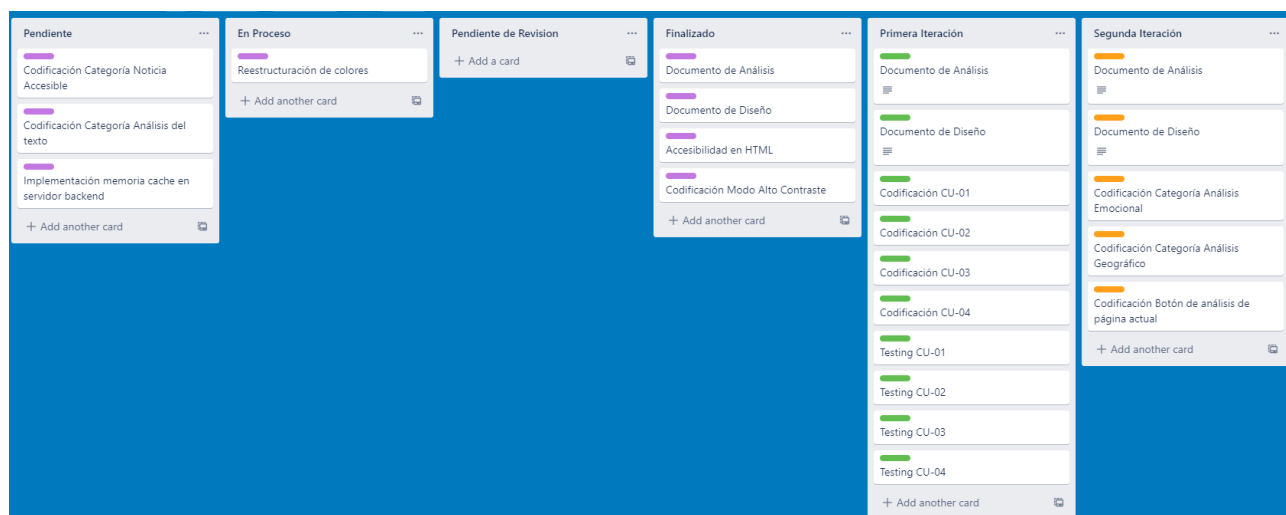


Figura B.26: Tablón de tareas de la iteración 3 - 4-3-20

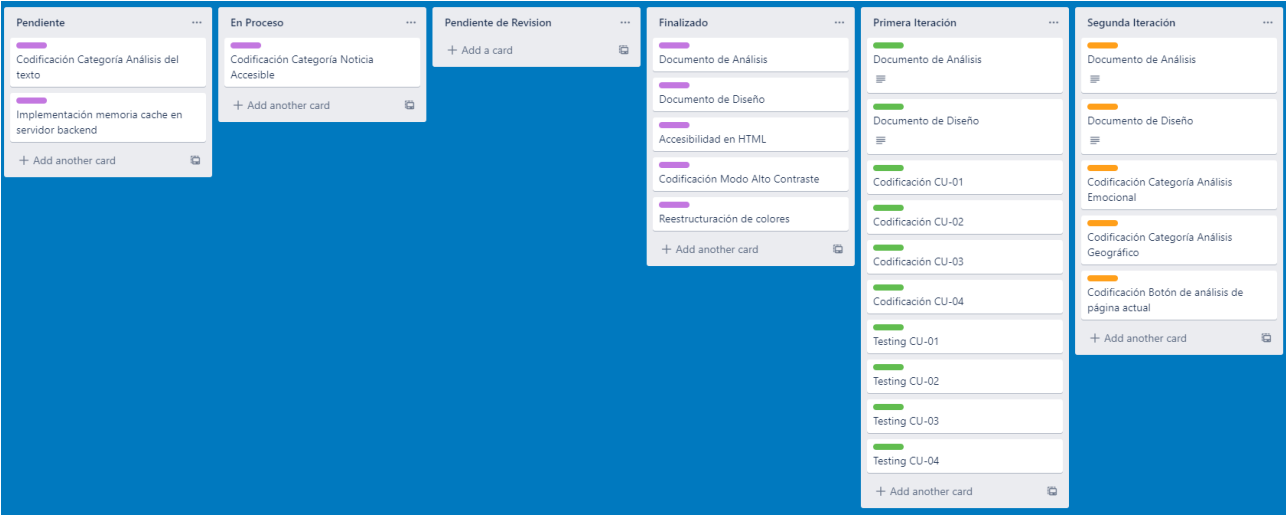


Figura B.27: Tablón de tareas de la iteración 3 - 7-3-20

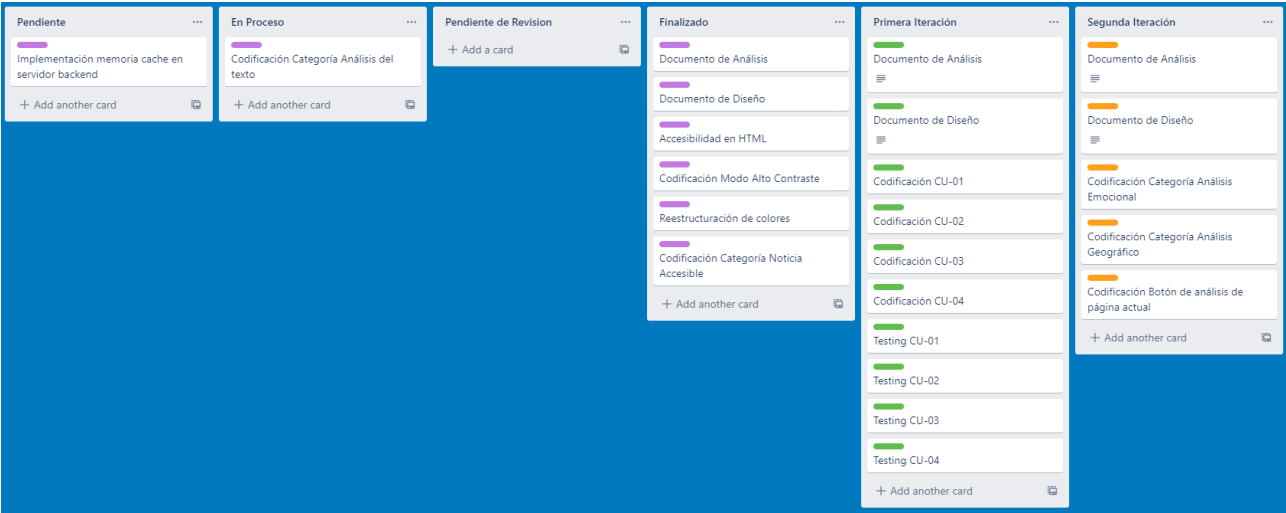


Figura B.28: Tablón de tareas de la iteración 3 - 8-3-20

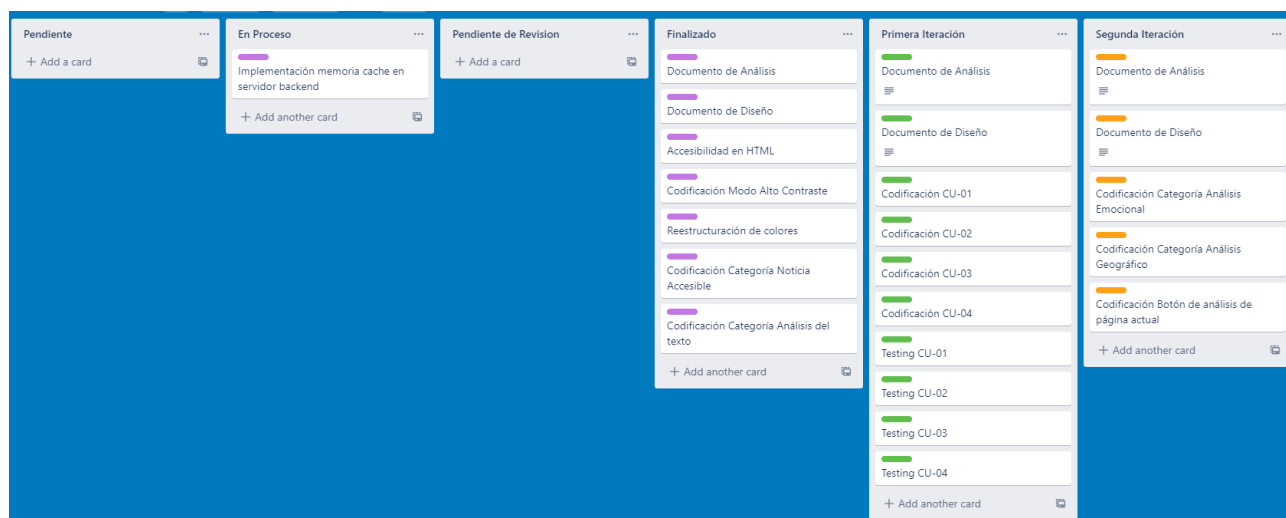


Figura B.29: Tablón de tareas de la iteración 3 - 10-3-20

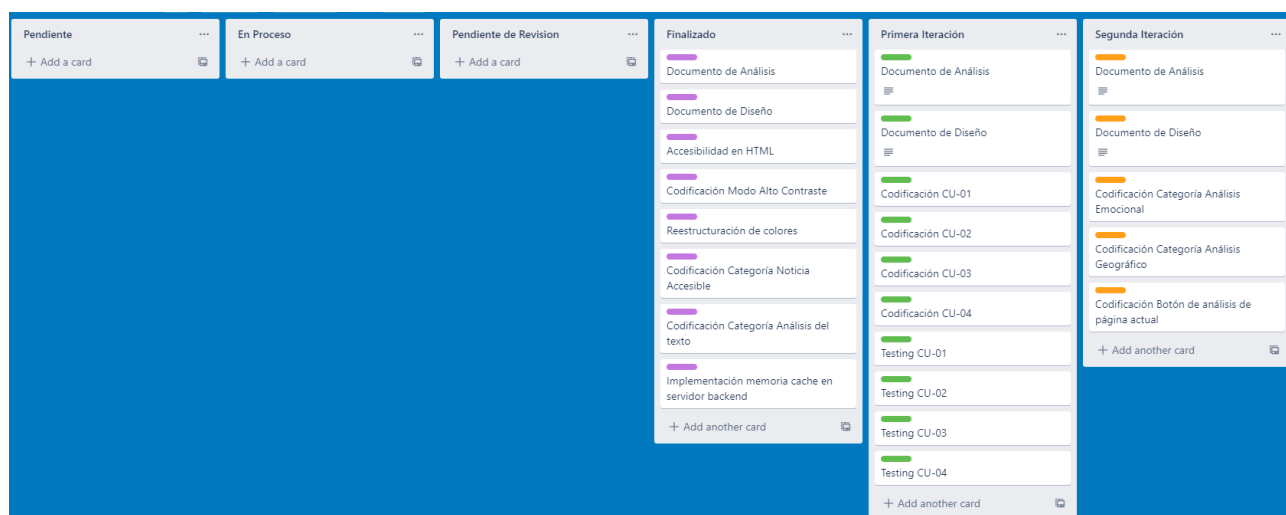


Figura B.30: Tablón de tareas de la iteración 3 - 12-3-20

ARQUITECTURA DE LA APLICACIÓN

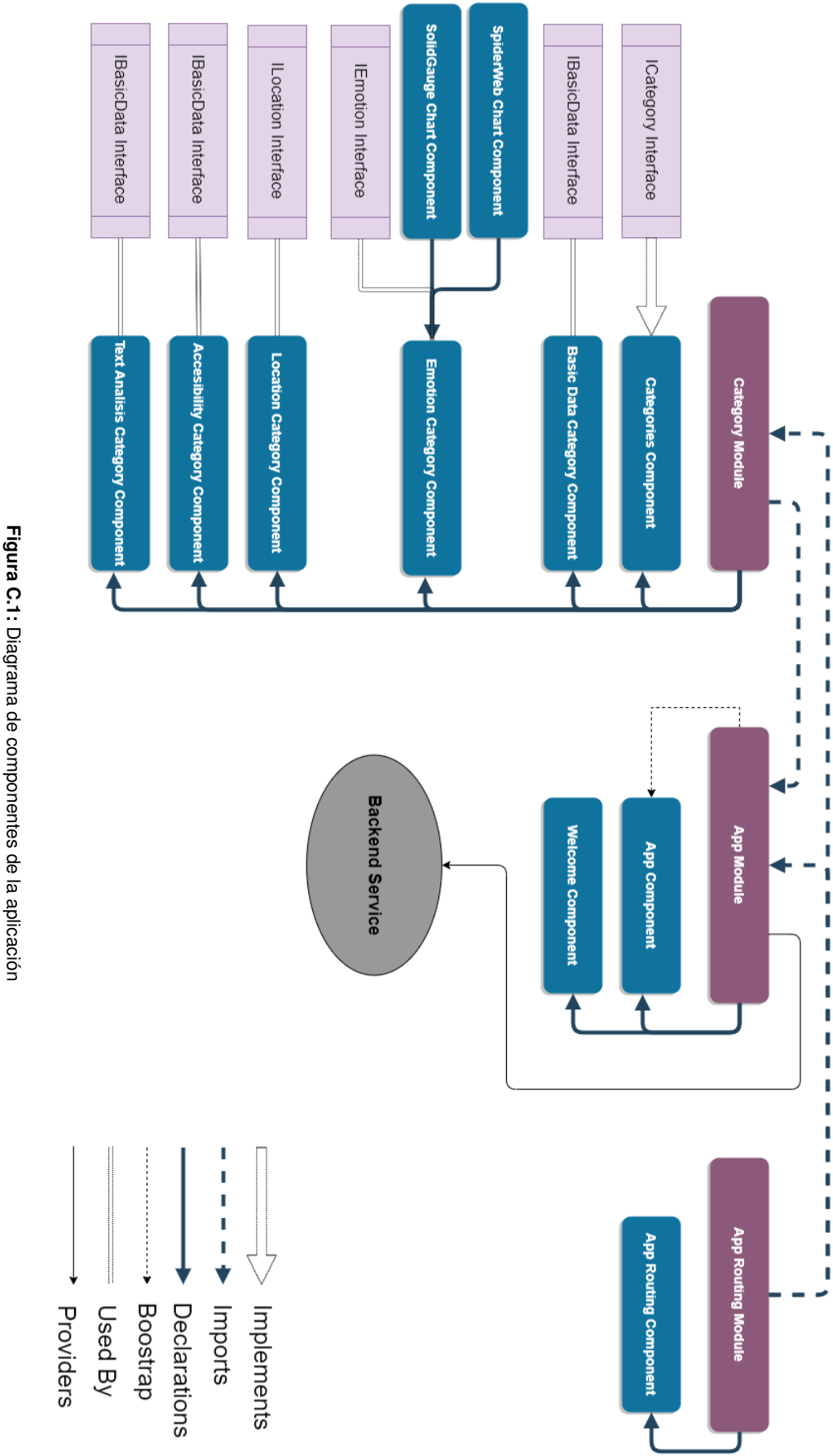


Figura C.1: Diagrama de componentes de la aplicación

MAQUETAS

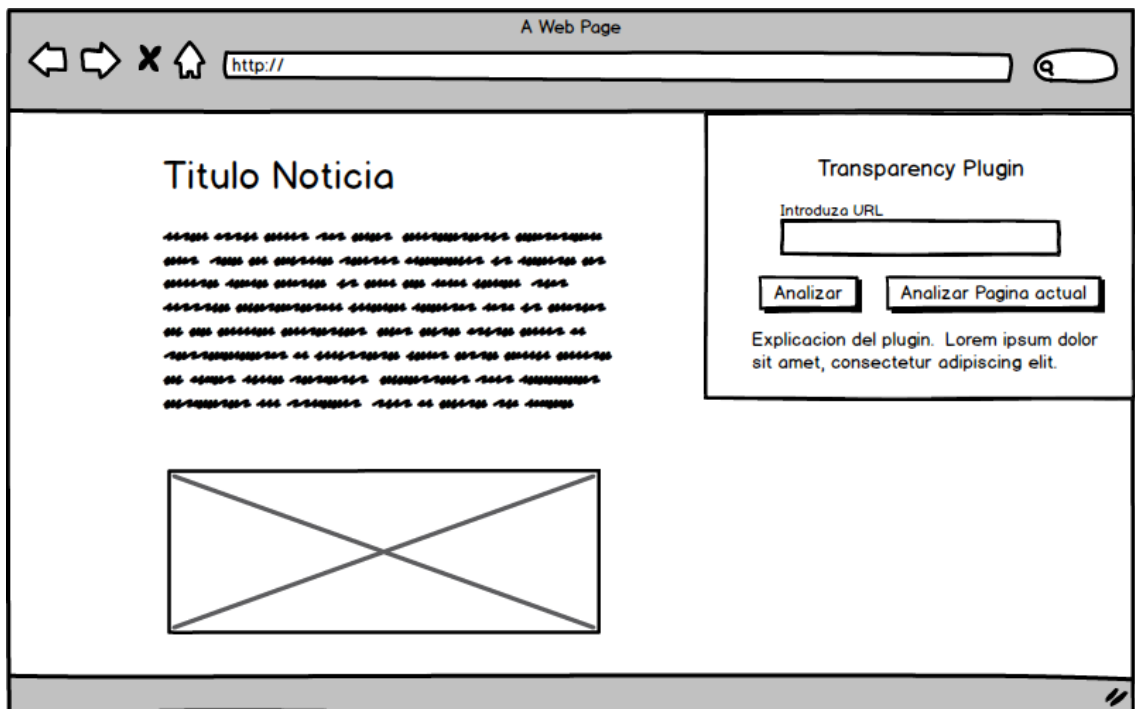


Figura D.1: Página de inicio de la aplicación



Figura D.2: Página de selección de categorías

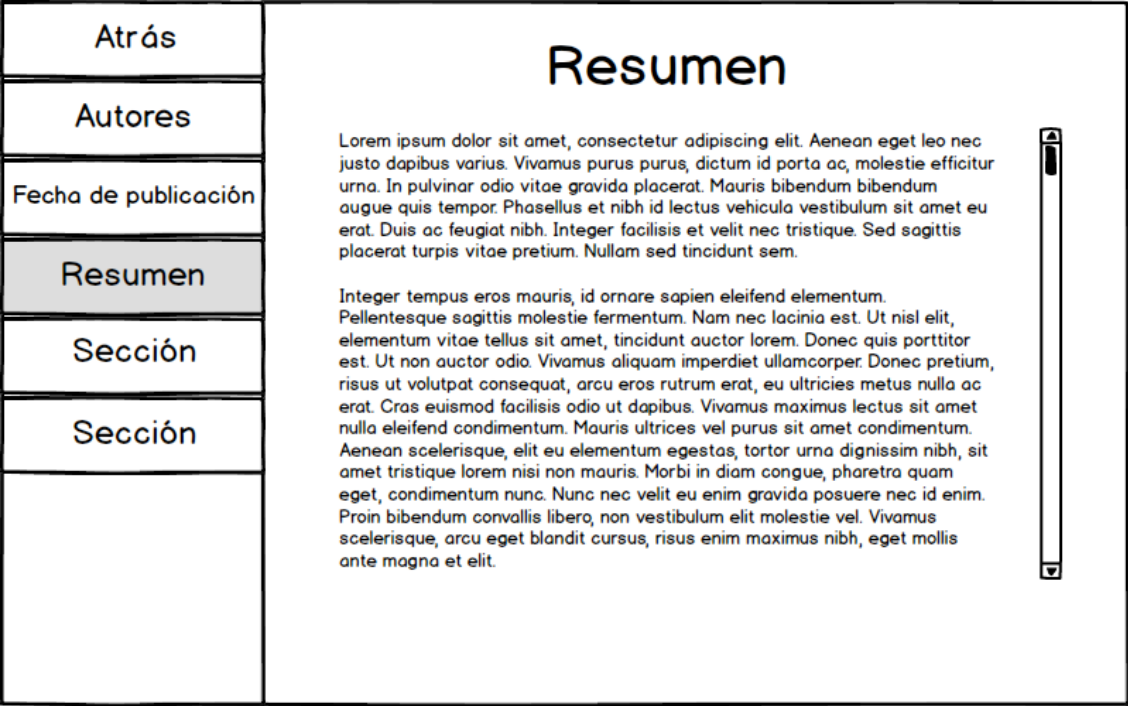


Figura D.3: Categoría de Datos Básicos

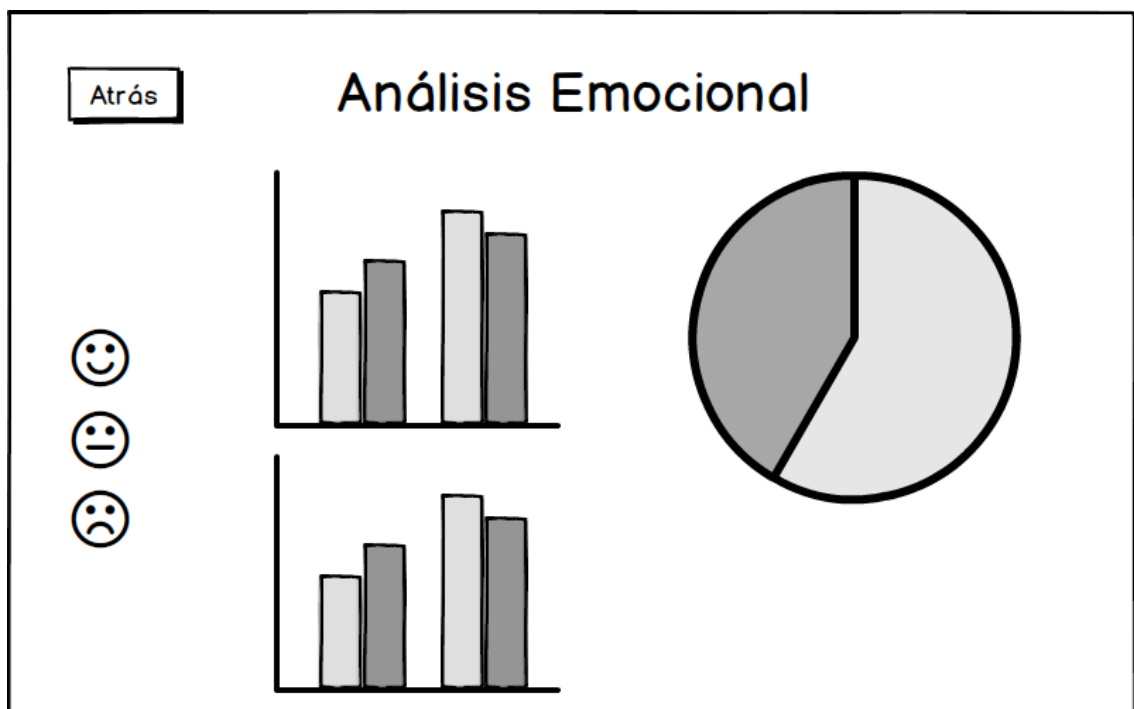


Figura D.4: Categoría de Análisis Emocional



Figura D.5: Categoría de Análisis Geográfico

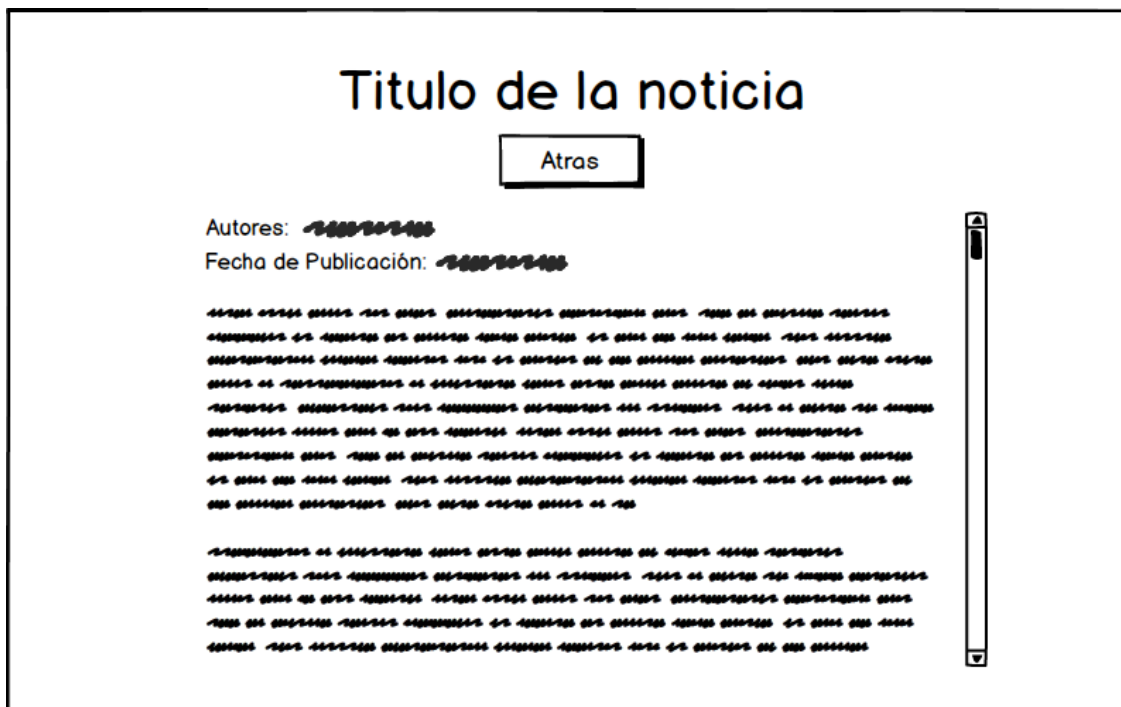


Figura D.6: Categoría de Noticia Accesible

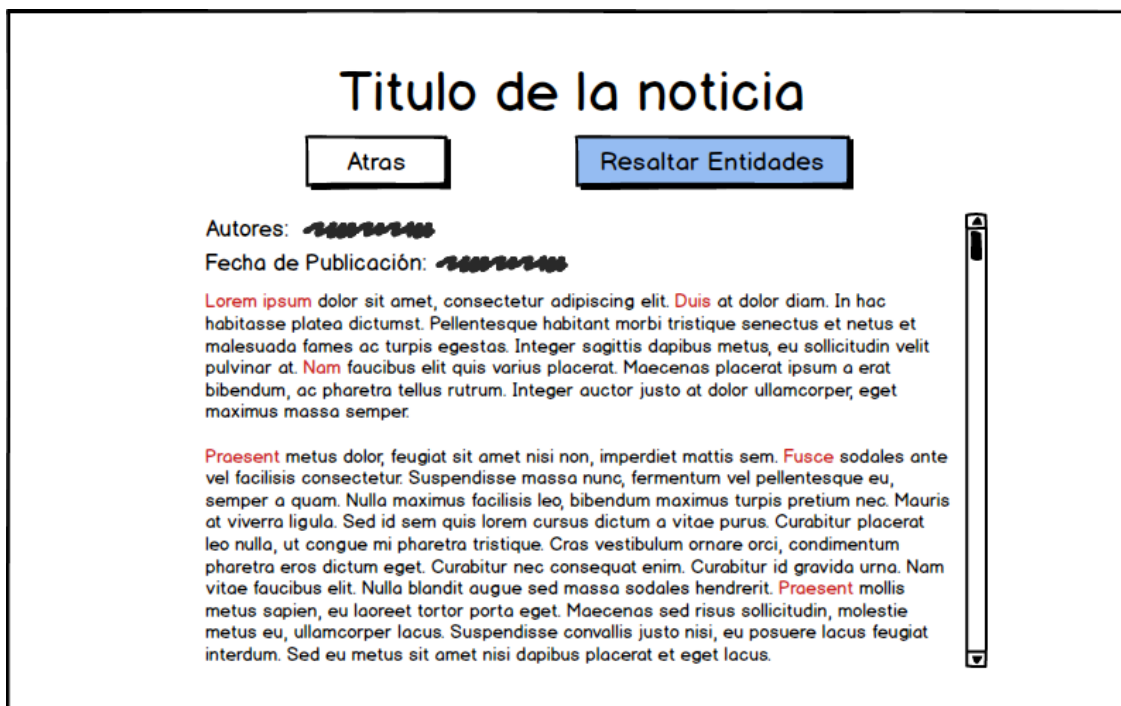


Figura D.7: Categoría de Análisis del Texto

APLICACIÓN FINAL

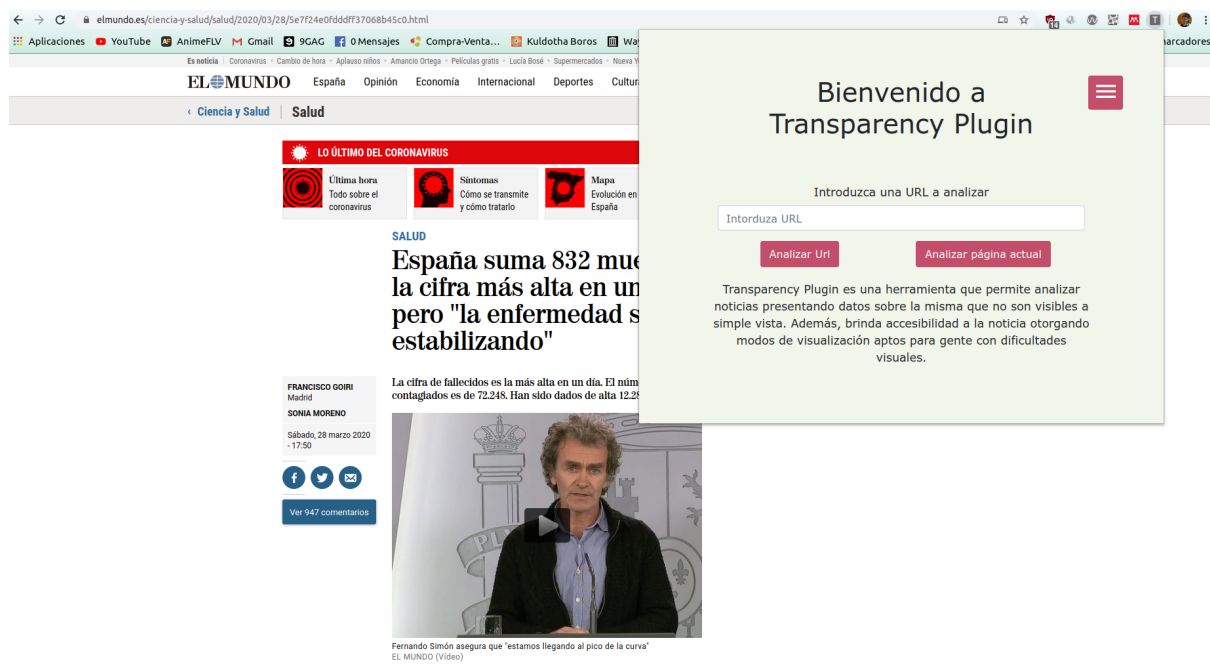


Figura E.1: Página de inicio de la aplicación



Figura E.2: Página de inicio de la aplicación



Figura E.3: Página de selección de categorías

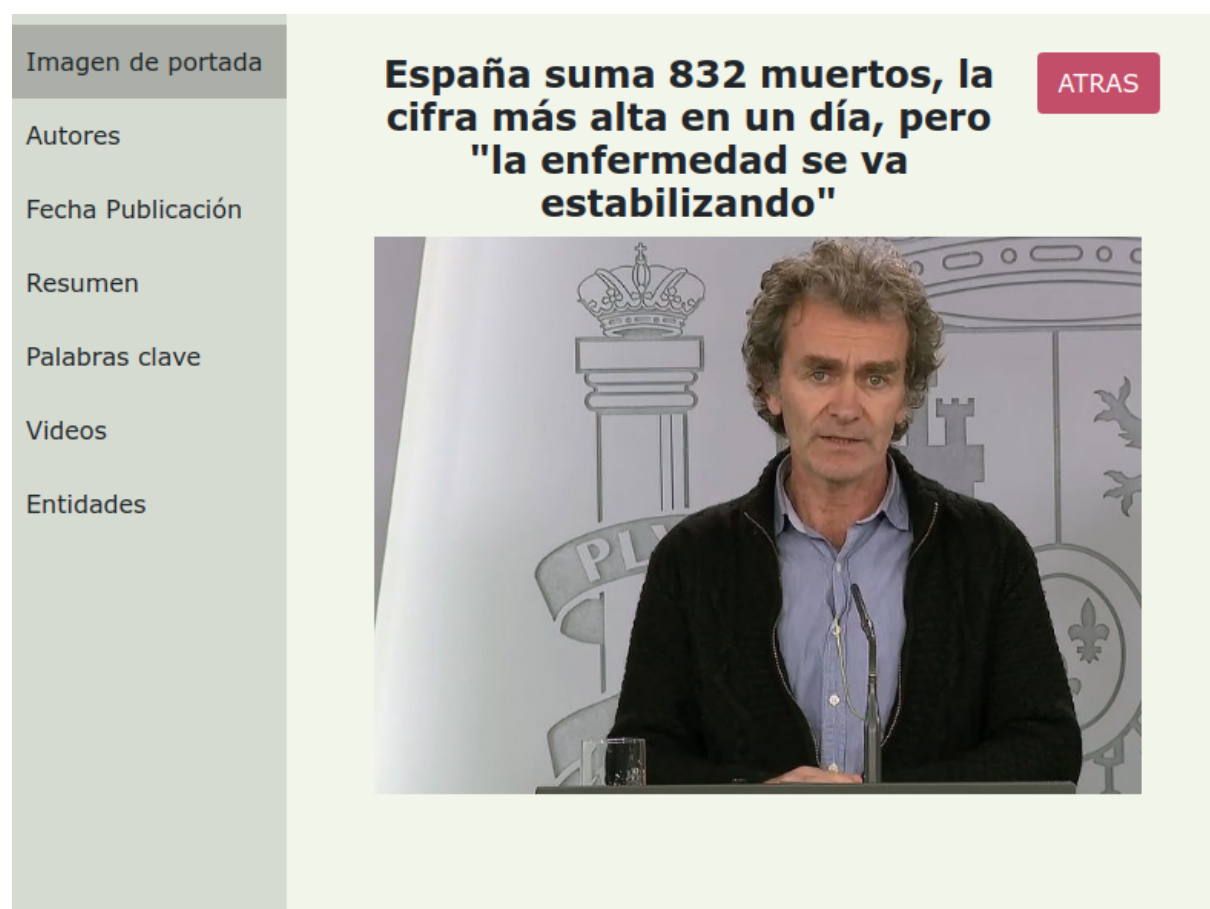


Figura E.4: Categoría de Datos Básicos - Parte 1

Imagen de portada	
Autores	
Fecha Publicación	
Resumen	<p>España suma 832 muertos, la cifra más alta en un día, pero "la enfermedad se va estabilizando"</p> <p>En un solo día ha habido en España casi tantos muertos por Covid-19 como fallecidos en todo 2019 por accidentes de tráfico (1.098). Es la cara y la cruz de la pandemia de coronavirus que azota España. Según el responsable ministerial, esa saturación "se ha alcanzado ya en dos comunidades y está a punto de alcanzarse en otras". Están enfocados a la fase preliminar de la enfermedad y buscan frenar la fase inmunológica. El ejército también está apoyando el traslado de los cadáveres a las diferentes morgues improvisadas en esta pandemia, y en el traslado de pacientes moderados entre hospitales de Madrid.</p>
Palabras clave	
Videos	
Entidades	

Figura E.5: Categoría de Datos Básicos - Parte 2

The screenshot displays a web application interface. On the left is a sidebar with the following navigation links: "Imagen de portada", "Autores", "Fecha Publicación", "Resumen", "Palabras clave", "Videos", and "Entidades". The "Entidades" link is currently selected and highlighted. The main content area features a headline: "España suma 832 muertos, la cifra más alta en un día, pero 'la enfermedad se va estabilizando'". To the right of the headline is a red button labeled "ATRÁS". Below the headline, there are three filterable sections: "Personas" (with a dropdown arrow), "Organizaciones" (with an upward arrow), and "Lugares" (with a dropdown arrow). The "Organizaciones" section is expanded, showing a list of five entities, each preceded by a black circle icon: "Ordenación Profesional Ministerio Sanidad", "Comité Gestión Técnica Coronavirus EI", "Centro Coordinación Alertas Emergencias", "UE", and "Agencia Española Medicamentos". At the bottom of the filterable sections is an "Otros" (Others) category with a dropdown arrow.

Figura E.6: Categoría de Datos Básicos - Parte 3

Análisis Geográfico

ATRÁS

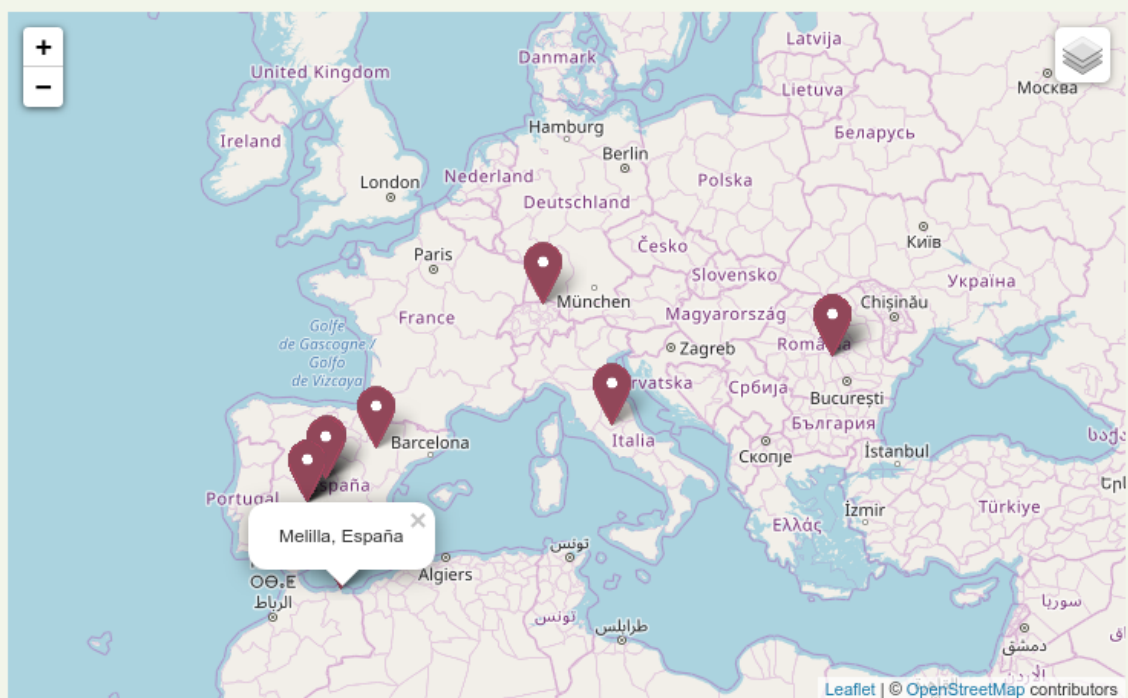


Figura E.7: Categoría de Análisis Geográfico

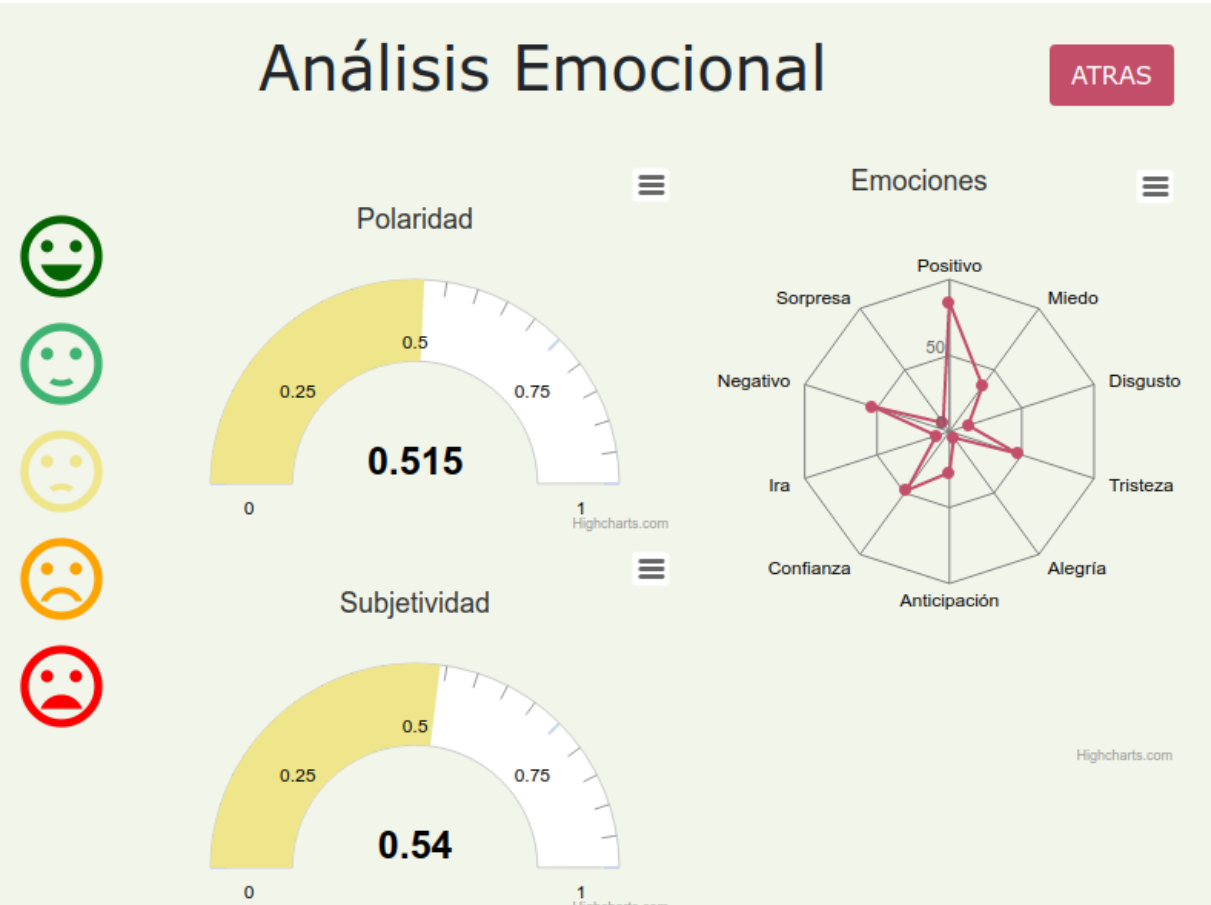


Figura E.8: Categoría de Análisis Emocional

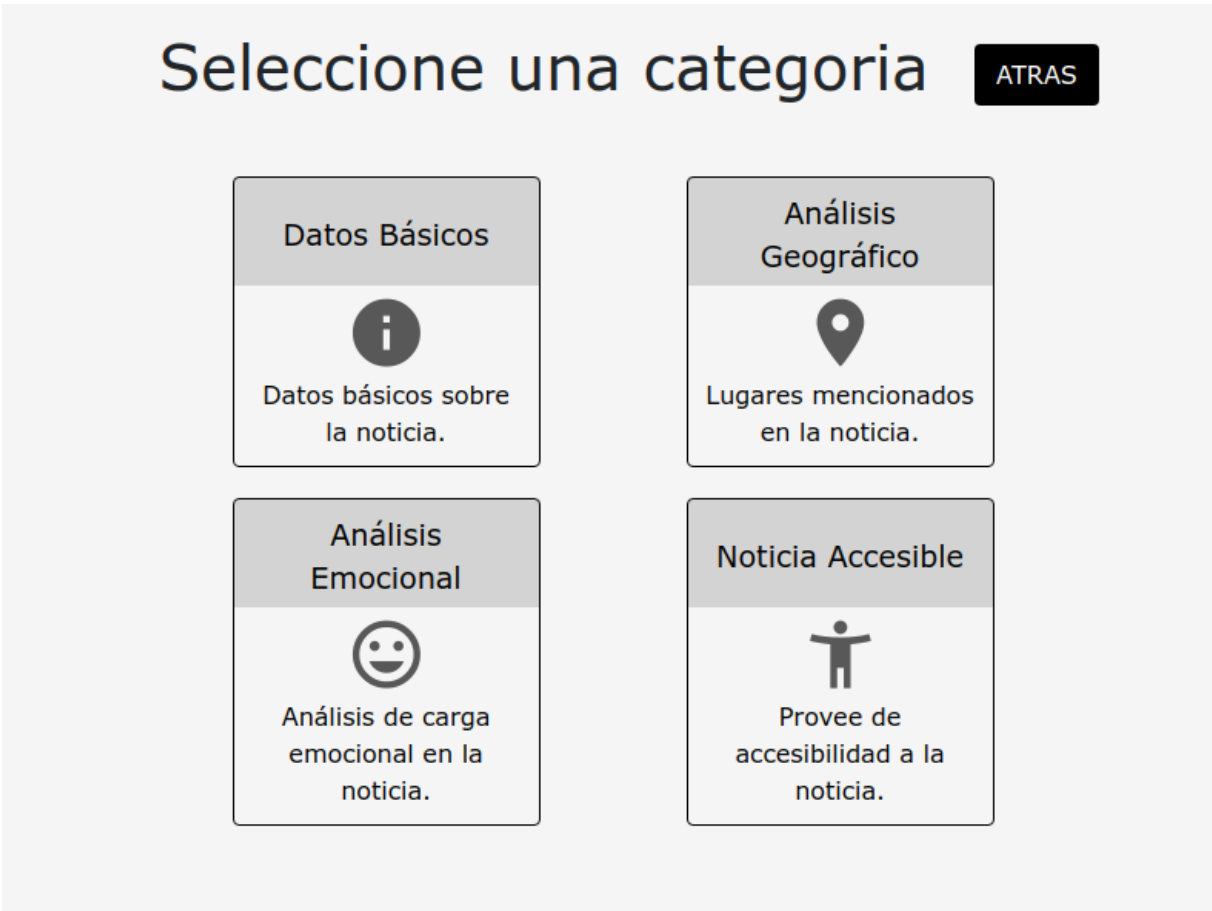


Figura E.9: Modo Alto Contraste - Parte 1

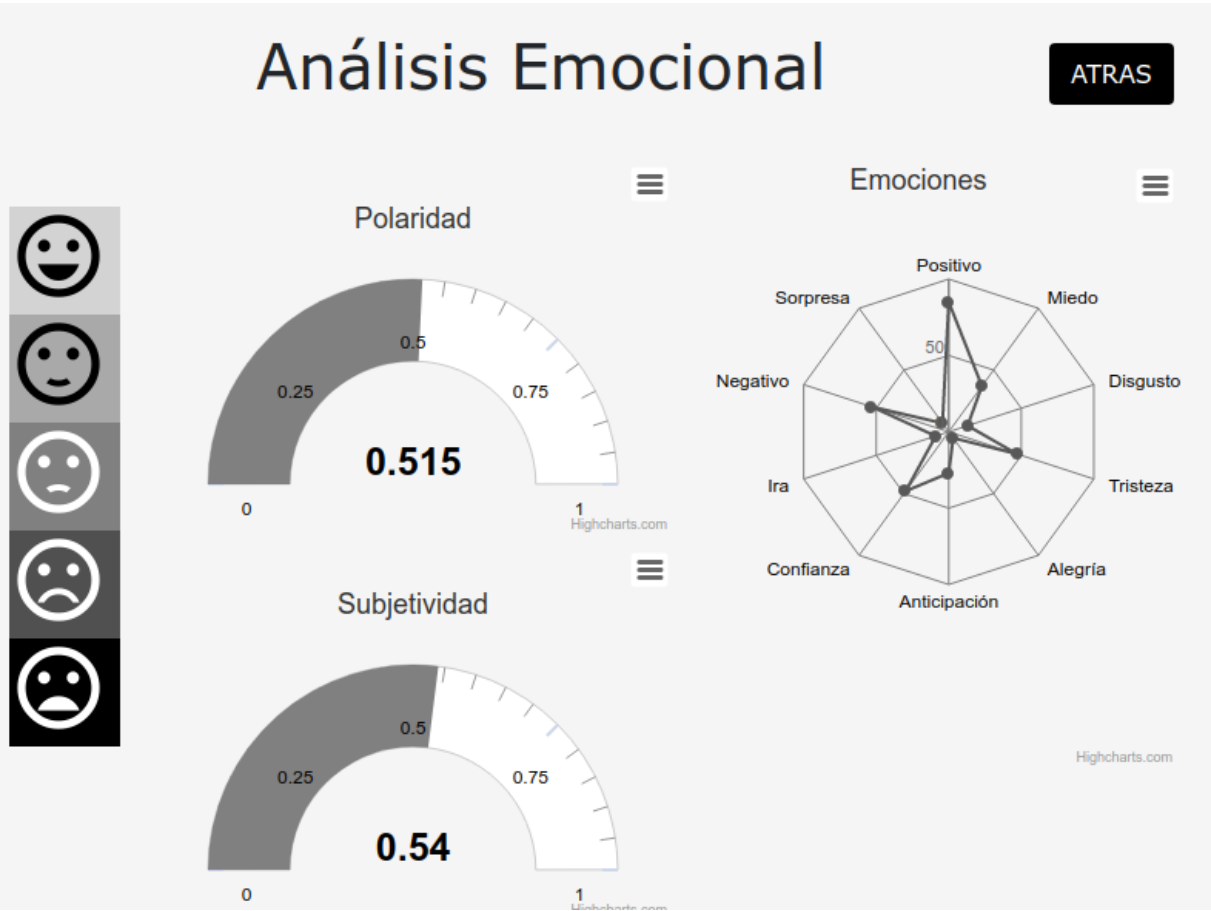


Figura E.10: Modo Alto Contraste - Parte 2

España suma 832 muertos, la cifra más alta en un día, pero "la enfermedad se va estabilizando"

ATRÁS

Autores: El Mundo,Vídeo,Francisco Goiri,Sonia Moreno

Fecha de publicación: 28/03/2020

La cifra de fallecidos es la más alta en un día. El número de contagiados es de 72.248. Han sido dados de alta 12.285 El dramático recuento de cifras que arroja a diario el coronavirus en España sigue siendo ascendente, tanto en número de casos como de fallecidos. Según los datos oficiales del Ministerio de Sanidad, España contabiliza hoy un total de 72.248 positivos confirmados, un 13% más que los registrados ayer, y los fallecidos ascienden ya a 5.690, frente a los 4.858 del viernes: si ayer el incremento de muertos con respecto al día anterior era de 769, hoy los datos oficiales hablan de un aumento de 832, la cifra más alta de fallecimientos en 24 horas desde que comenzó la pandemia. En un solo día ha habido en España casi tantos muertos por Covid-19 como fallecidos en todo 2019 por accidentes de tráfico (1.098). El número de fallecidos por Covid-19 se ha multiplicado por 162 en menos de tres semanas, mientras que el número

Figura E.11: Categoría de Noticia Accesible

España suma 832 muertos, la cifra más alta en un día, pero "la enfermedad se va estabilizando"

ATRAS

No Resaltar Entidades

demuestra que muchos casos entre los profesionales han sido leve". Ha añadido, además, que el elevado número de casos detectados en este colectivo puede leerse además como resultado del "enorme esfuerzo que se está haciendo para detectar todos los casos confirmados". **Rodrigo Gutiérrez** PERSON, director general de Ordenación

Profesional del Ministerio de Sanidad, ha hecho un recuento de los profesionales incorporados en los últimos días al sistema para el combate contra la enfermedad, y ahí ha incluido a los profesionales que hicieron el **MIR** MISC en la convocatoria 2018-2019,

a los extranjeros, tanto de fuera como de dentro de la **UE** ORG (aunque aún no tengan

la homologación definitiva del título), a médicos y enfermeros jubilados, personal emérito y liberados sindicales, así como estudiantes de último año de Medicina y

Enfermería en labores de apoyo. **Además** LOC, el Ministerio de Justicia ha puesto a

Figura E.12: Categoría de Análisis del Texto

INFORME DE CONSENTIMIENTO DE DATOS

Usted va a participar en un estudio empírico llevado a cabo por el alumno Germán Darío Esquinazi Bachoer para evaluar la usabilidad del plugin Transparency Plugin. Realizará una tarea guiada en la que se le pedirá que emplee un plugin para el navegador Chrome y localice e interprete cierta información específica. Tras esto, deberá rellenar un cuestionario.

Para garantizar el buen funcionamiento de este experimento y proteger plenamente sus derechos e intereses, antes de aceptar participar, debe conocer claramente la siguiente información:

- Esta tarea no tendrán ninguna repercusión en la calificación de las asignaturas que esté cursando.
- Durante la realización del estudio, llevará a cabo una tarea guiada consistente de 7 pasos que le pedirán que busque cierta información presente en el plugin.
- Su participación en el experimento es completamente voluntaria.
- Usted podrá cancelar su participación en la investigación en cualquier momento, sin sufrir penalizaciones por esta decisión.

Aseguramos que los datos o el material se utilizarán únicamente con fines educativos y de investigación. Si comprende y acepta completamente lo anterior, por favor rellene este consentimiento informado.

Fecha

Acepto las condiciones

CUESTIONARIO DE USABILIDAD

G.1. Preguntas destinadas al cálculo del SUS

Creo que me gustaría usar esta herramienta con frecuencia *

Encontré esta herramienta innecesariamente compleja *

Creo que la herramienta es fácil de usar *

Creo que necesitaría ayuda para poder usar esta herramienta *

He encontrado que las diversas funciones de esta herramienta estaban bien integradas *

Creo que hay demasiadas funciones inconsistentes en esta herramienta *

Creo que la mayoría de la gente puede aprender a usar esta herramienta muy rápidamente *

He encontrado esta herramienta muy engorrosa/incómoda de usar *

Me sentí muy seguro de lo que hacía al usar esta herramienta *

Tengo que aprender un montón de cosas antes de poder usar esta herramienta *

Figura G.1: Parte general del cuestionario para usuarios

G.2. Preguntas destinadas a medir la satisfacción

Sección 2 de 2

Cuestionario específico del plugin

En la categoría de análisis emocional, comprendo la información que me quieren transmitir los gráficos. *

Las dimensiones del plugin resultan inapropiadas en relación con la página a analizar. *

El plugin tarda poco al responder a las peticiones. *

Los colores usados dificultan o molestan el uso del plugin. *

Creo que puede resultar útil la opción de alto contraste para facilitar el acceso a la información a personas con problemas de accesibilidad *

Existen textos de categoría que no se corresponden con lo que hace esta misma. *

El plugin tiene un tamaño correcto con respecto a la página web. *

Los tiempos de respuesta del plugin resultan lentos. *

Figura G.2: Parte específica del cuestionario para usuarios - Primera mitad

Encuentro agradables los colores elegidos en la interfaz del plugin. *

...

Resulta complicado entender el propósito de los gráficos que se encuentran en la categoría de análisis emocional. *

En la categoría de análisis del texto, los tipos de entidades son los adecuados. *

La opción de alto contraste dificulta el entendimiento de la información presentada por el plugin. *

Encuentro el botón de volver atrás en una posición agradable e intuitiva. *

Las entidades que aparecen en el análisis del texto, son insuficientes y deberían añadirse o reemplazarse por otros. *

El nombre, texto e icono de cada categoría me ha ayudado a entender el propósito de esta. *

El botón para retroceder a la pantalla anterior está es difícil de encontrar. *

Figura G.3: Parte específica del cuestionario para usuarios - Segunda mitad

RESULTADOS DEL CUESTIONARIO DE USABILIDAD

H.1. Respuestas para el SUS

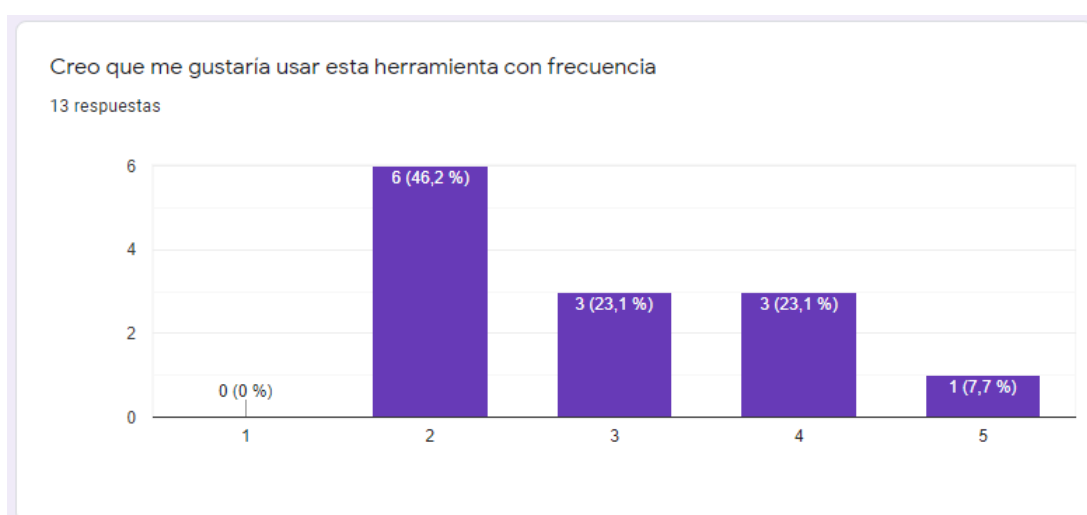


Figura H.1: Resultado de la pregunta 1 del cuestionario de usabilidad

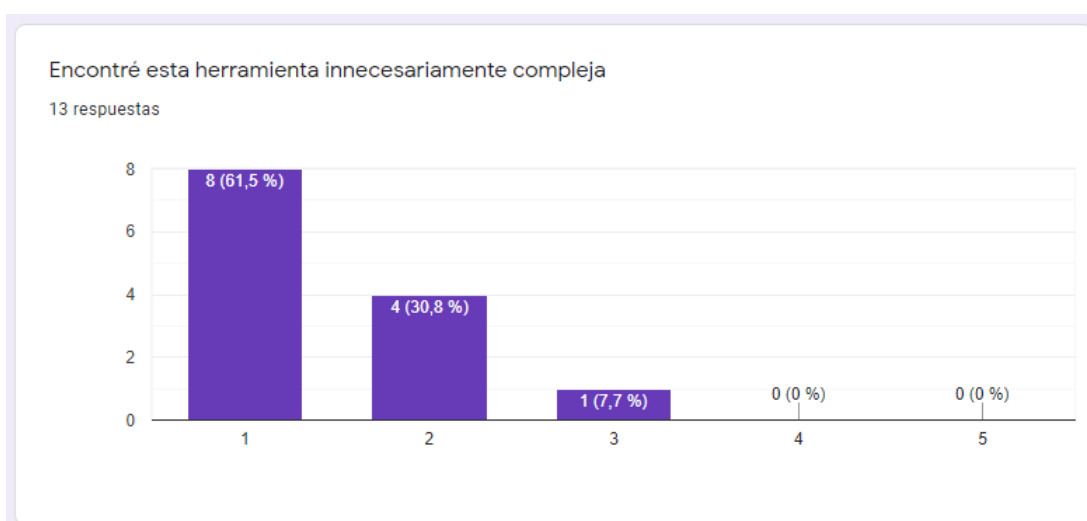


Figura H.2: Resultado de la pregunta 2 del cuestionario de usabilidad

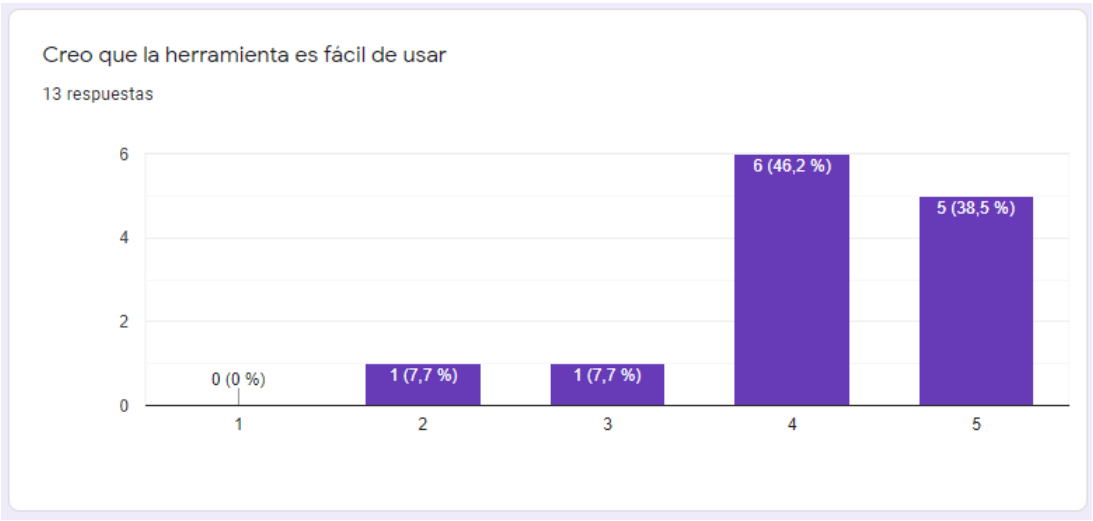


Figura H.3: Resultado de la pregunta 3 del cuestionario de usabilidad

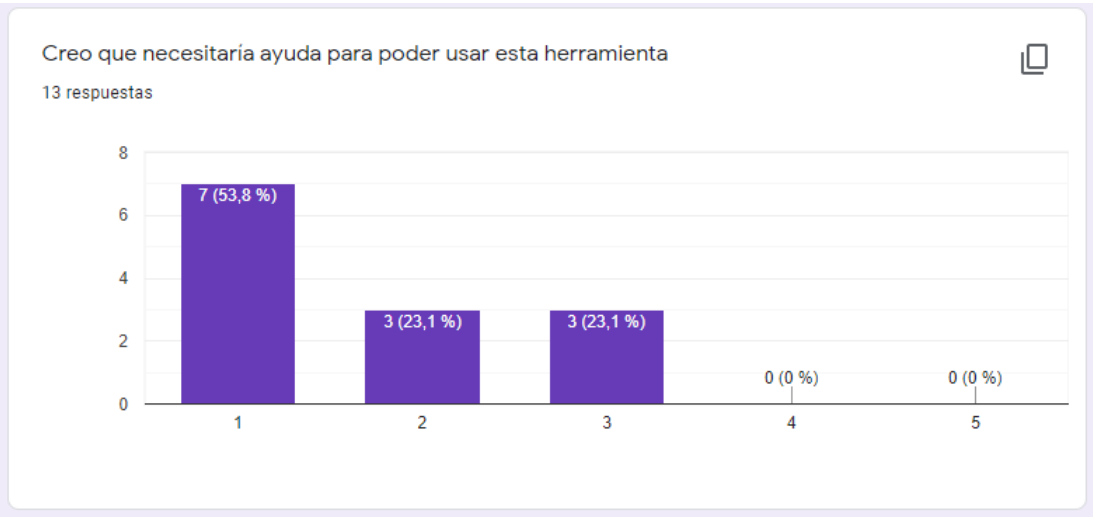


Figura H.4: Resultado de la pregunta 4 del cuestionario de usabilidad

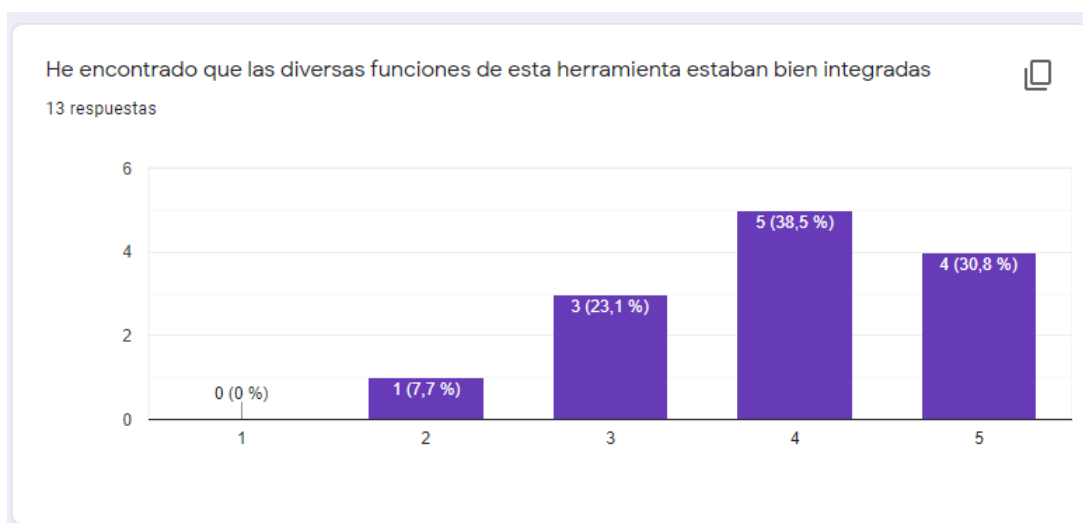


Figura H.5: Resultado de la pregunta 5 del cuestionario de usabilidad

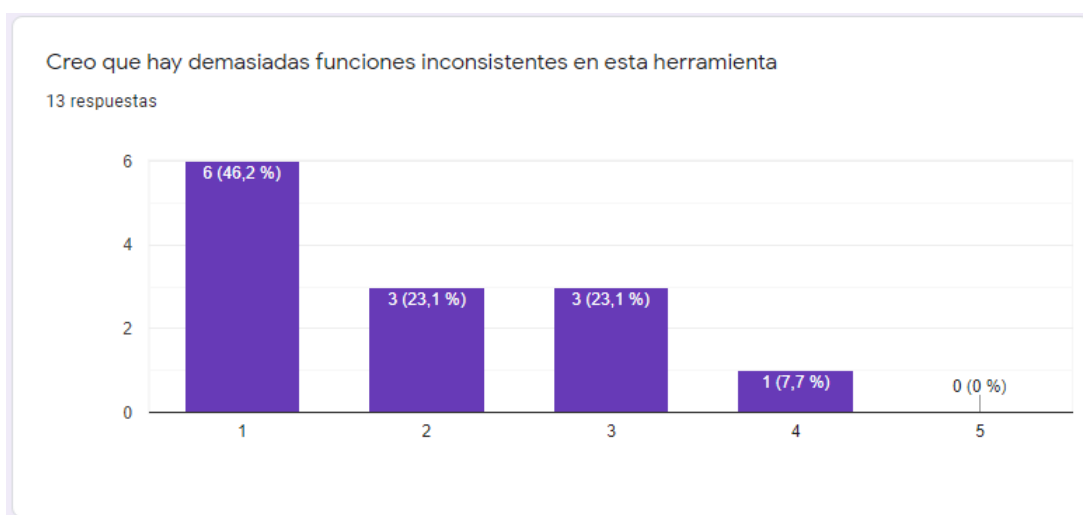


Figura H.6: Resultado de la pregunta 6 del cuestionario de usabilidad

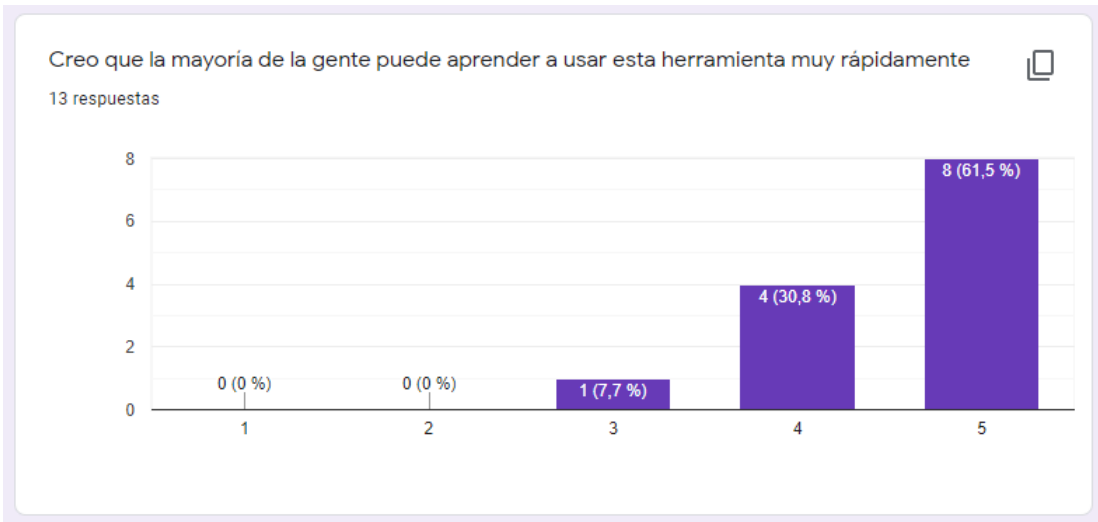


Figura H.7: Resultado de la pregunta 7 del cuestionario de usabilidad

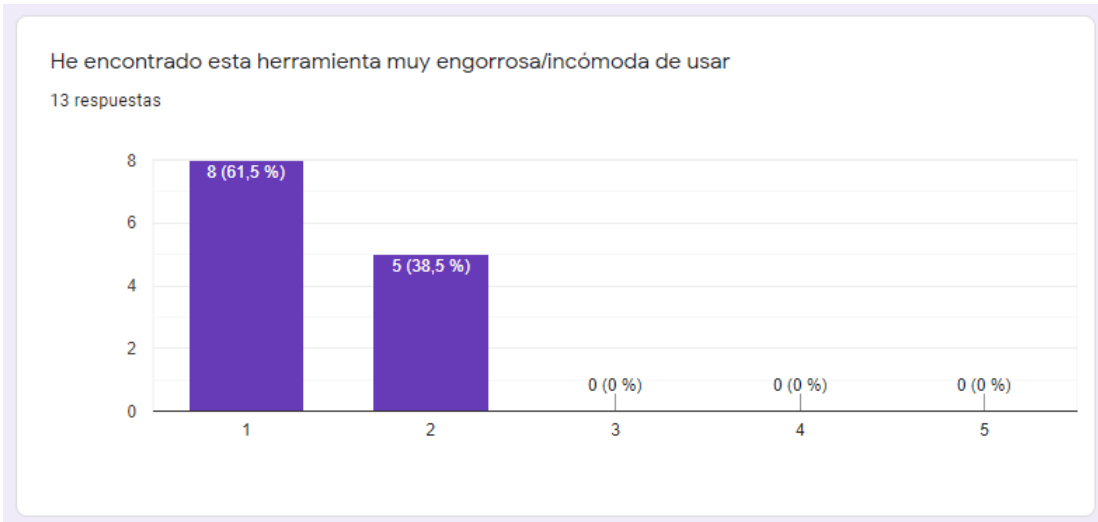


Figura H.8: Resultado de la pregunta 8 del cuestionario de usabilidad



Figura H.9: Resultado de la pregunta 9 del cuestionario de usabilidad

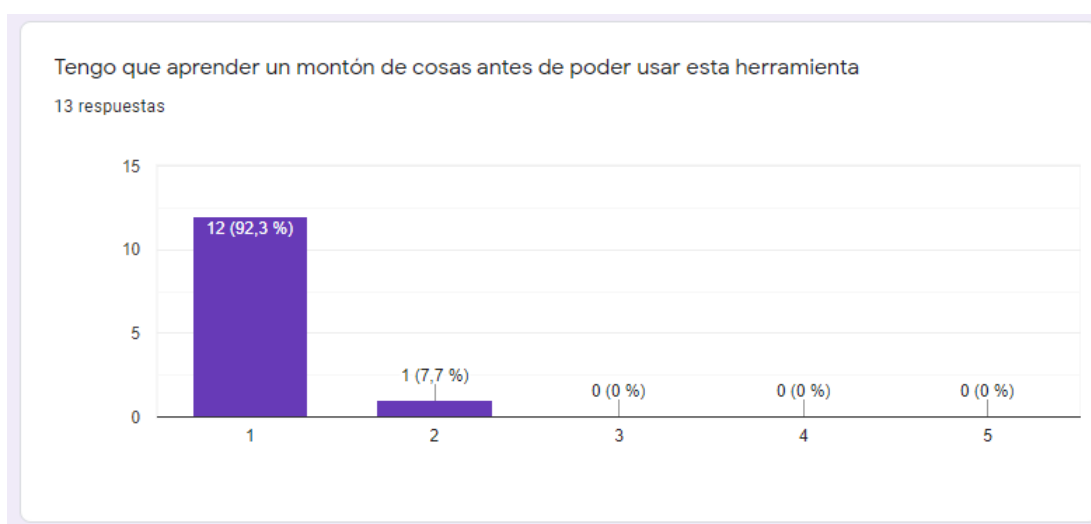


Figura H.10: Resultado de la pregunta 10 del cuestionario de usabilidad

H.2. Respuestas para la satisfacción

Cuestionario específico del plugin

Describe con pocas palabras que representa cada gráfico en la categoría de Análisis Emocional (Polaridad, Subjetividad y Emociones)

13 respuestas

Polaridad: cuánto destaca una emoción sobre el resto de emociones en la página
 Subjetividad: cantidad de opinión personal que contiene la página
 Emociones: proporción de cada uno de los sentimientos en la página en base a las palabras utilizadas

Polaridad: posición siendo el 0/rojo lo negativo y el 1/verde lo positivo
 Subjetividad: lo objetiva (0) o subjetiva (1) que es la noticia; no sé lo que es el color
 Emociones: qué emociones predominan y con cuántas palabras aparecen en el texto

He entendido que la polaridad indica si el autor esta a favor o en contra del tema que trata en la noticia y la subjetividad si está siendo parcial o imparcial con el tema tratado. El que más me ha gustado es el de las emociones porque da una imagen de lo que la noticia transmite a través de las palabras que utiliza. Quizá encuentre algo poco intuitivo el tipo de gráficos usados para subjetividad y polaridad pero más o menos se entienden.

Polaridad=carga política,Subjetividad=lo que la propia palabra indica, emociones=carga emocional de la noticia

Polaridad: Radicalización del pensamiento de la noticia
 Subjetividad: Neutralidad a la hora de narrar la noticia

Figura H.11: Resultado de la pregunta 11 del cuestionario de usabilidad - Parte 1

Cuestionario específico del plugin

Describe con pocas palabras que representa cada gráfico en la categoría de Análisis Emocional (Polaridad, Subjetividad y Emociones)

13 respuestas

Polaridad: Radicalización del pensamiento de la noticia
Subjetividad: Neutralidad a la hora de narrar la noticia
Emociones: Establece de manera cuantitativa la carga emocional de la noticia

Polaridad: Describe la inclinación ideológica de la noticia
Subjetividad: Describe la cantidad de opinión personal que da el autor/es
Emociones: La intencionalidad o resultado que transmite la noticia al lector

La polaridad no llego a entender lo que significa

La subjetividad mide el grado de subjetividad del autor a la hora de escribir la noticia

La gráfica de emociones indica como de presente están reflejados los sentimientos en la noticia.

polaridad: tendencia de la noticia. Subjetividad: como esta interpretada la noticia Emociones: lo que provoca en el lector la lectura de la noticia

polaridad no me representa nada y subjetividad tampoco, la parte de emociones si lo veo mas acorde al titulo

Figura H.12: Resultado de la pregunta 11 del cuestionario de usabilidad - Parte 2

Cuestionario específico del plugin

Describe con pocas palabras que representa cada gráfico en la categoría de Análisis Emocional (Polaridad, Subjetividad y Emociones)

13 respuestas

polaridad no me representa nada y subjetividad tampoco,la parte de emociones si lo veo mas acorde al titulo

el de pariad no lo entiendo ,la de subjetividad es muy grafica y las de emociones esta tambien muy bien que mide graficamente lo que inspira emocionalmente la notica

Polaridad, al principio no esta muy claro lo que describe. Mi idea inicial fue que describia si el articulo trata de convencerte de algo. Por ejemplo un articulo con ideas muy de derechas o de izquierdas, es un articulo muy polarizado. En cambio un articulo que da opiniones diversas sobre el tema no esta polarizado. Luego de pensarlo un poco, creo que el termino polaridad viene de polo positivo/negativo. Es decir, si la noticia transmite una emocion positiva o negativa, lo cual concuerda con el titulo de la categoria "Analisis Emocional".

Subjetividad, describe si en el articulo hay muchas opiniones personales (muy subjetivo) o si los datos presentados son mas bien datos empiricos (poco subjetivo, objetivo).

El grafico de emociones indica la distribucion de las emociones del texto, hacia el centro poco de esa emocion, hacia los extremos, mucho de esa emocion.

Figura H.13: Resultado de la pregunta 11 del cuestionario de usabilidad - Parte 3

Cuestionario específico del plugin

Describe con pocas palabras que representa cada gráfico en la categoría de Análisis Emocional (Polaridad, Subjetividad y Emociones)

13 respuestas

de convencerte de algo. Por ejemplo un artículo con ideas muy de derechas o de izquierdas, es un artículo muy polarizado. En cambio un artículo que da opiniones diversas sobre el tema no está polarizado. Luego de pensarlo un poco, creo que el término polaridad viene de polo positivo/negativo. Es decir, si la noticia transmite una emoción positiva o negativa, lo cual concuerda con el título de la categoría "Análisis Emocional".

Subjetividad, describe si en el artículo hay muchas opiniones personales (muy subjetivo) o si los datos presentados son más bien datos empíricos (poco subjetivo, objetivo).

El gráfico de emociones indica la distribución de las emociones del texto, hacia el centro poco de esa emoción, hacia los extremos, mucho de esa emoción.

Polaridad: nivel de positividad/negatividad de la noticia.
Emociones: análisis de la carga emocional dividido en varias emociones.
Subjetividad: parcialidad de la noticia.

polaridad: ni idea (afinidad política?)
subjetividad: 0 totalmente objetivo y 1 totalmente subjetivo
emociones: tono que transmite el texto a nivel emocional

Figura H.14: Resultado de la pregunta 11 del cuestionario de usabilidad - Parte 4



Figura H.15: Resultado de la pregunta 12 del cuestionario de usabilidad

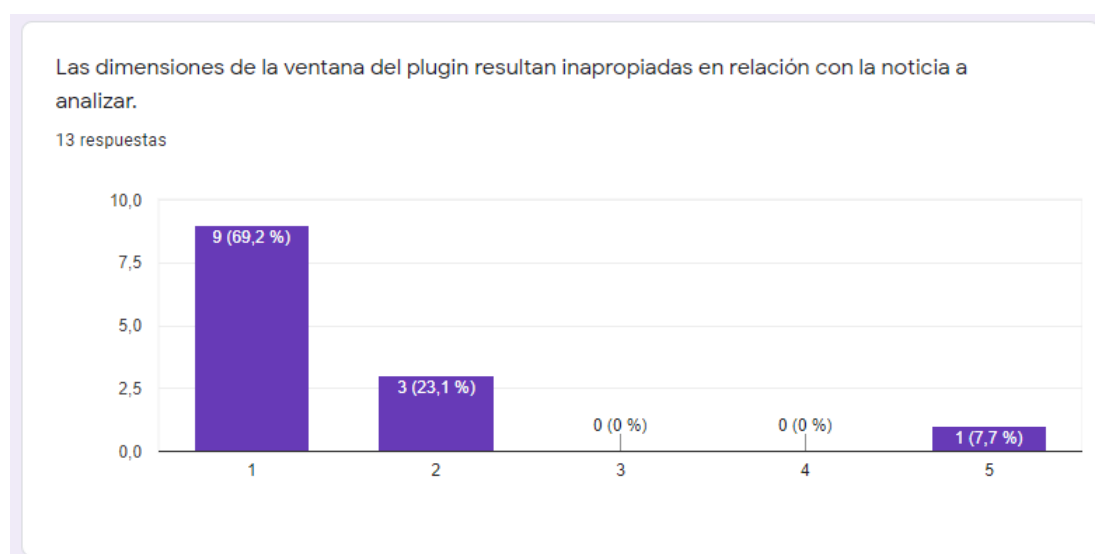


Figura H.16: Resultado de la pregunta 13 del cuestionario de usabilidad

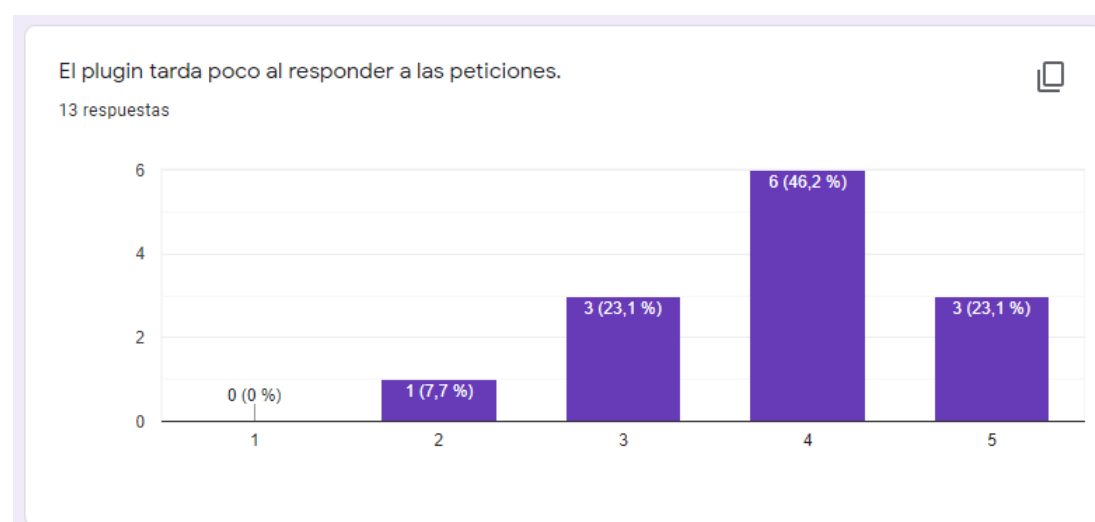


Figura H.17: Resultado de la pregunta 14 del cuestionario de usabilidad

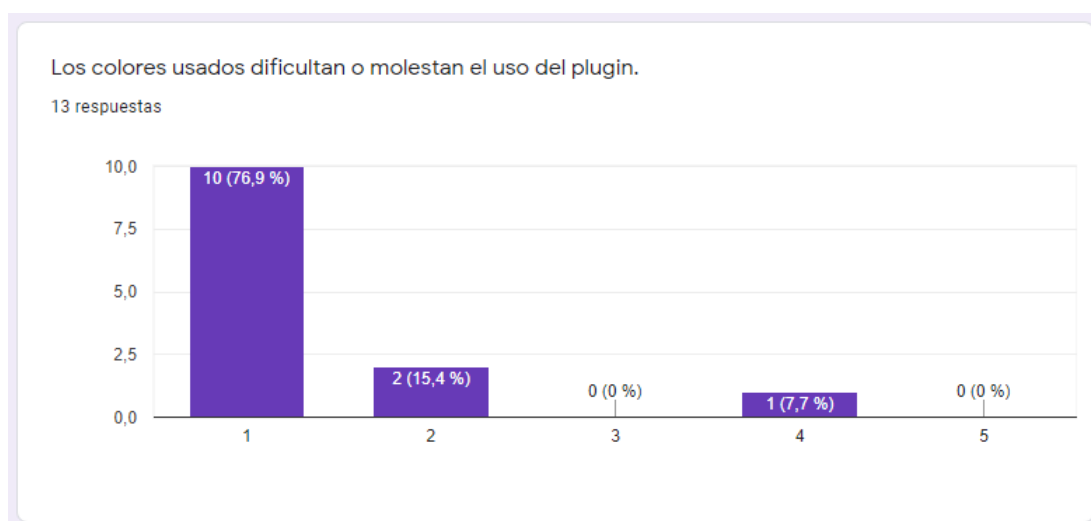


Figura H.18: Resultado de la pregunta 15 del cuestionario de usabilidad

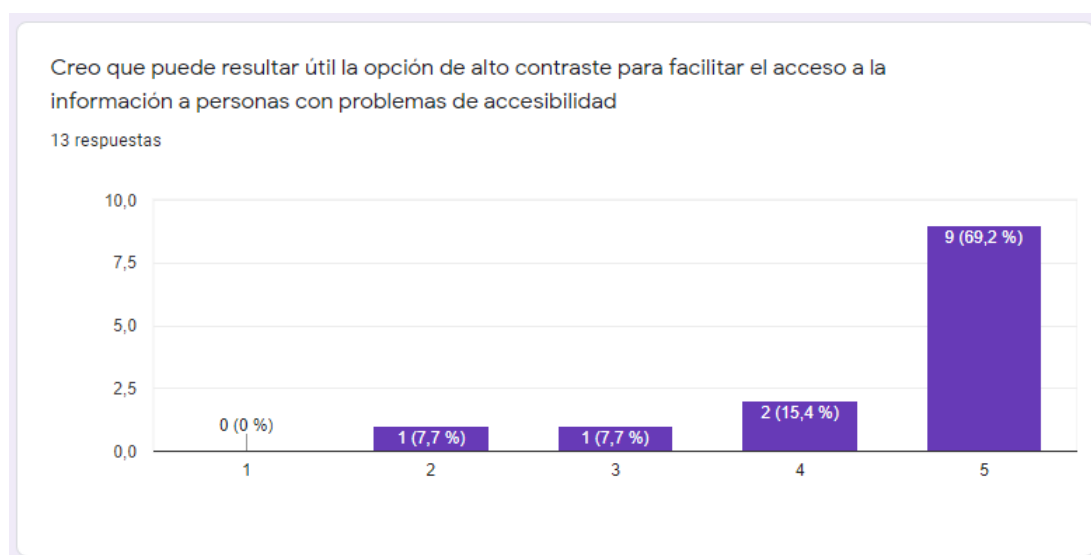


Figura H.19: Resultado de la pregunta 16 del cuestionario de usabilidad



Figura H.20: Resultado de la pregunta 17 del cuestionario de usabilidad



Figura H.21: Resultado de la pregunta 18 del cuestionario de usabilidad

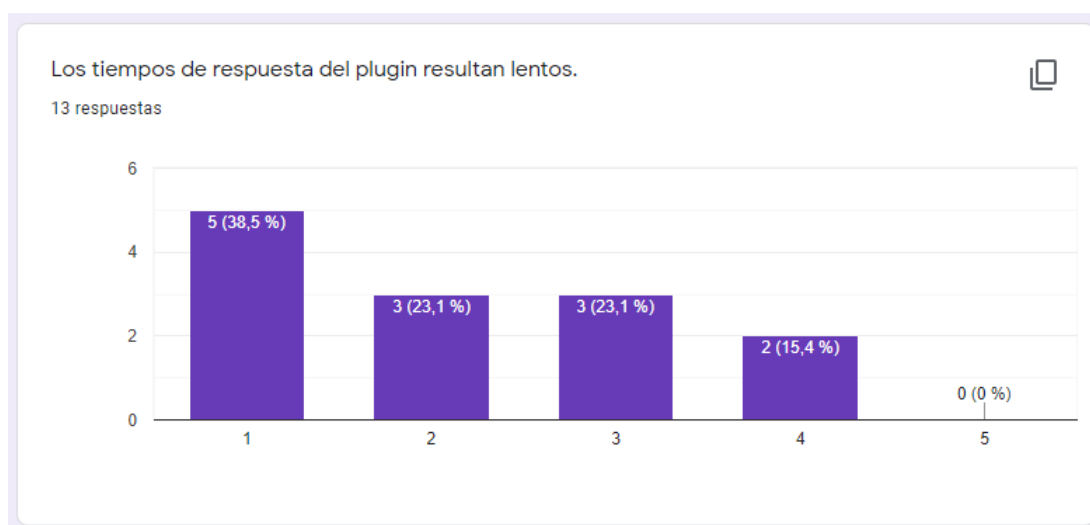


Figura H.22: Resultado de la pregunta 19 del cuestionario de usabilidad

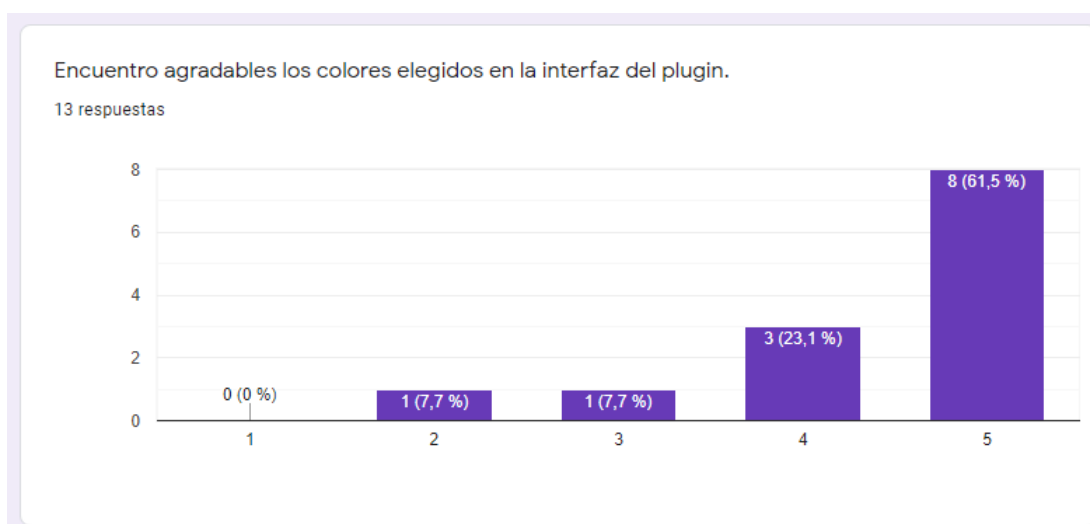


Figura H.23: Resultado de la pregunta 20 del cuestionario de usabilidad

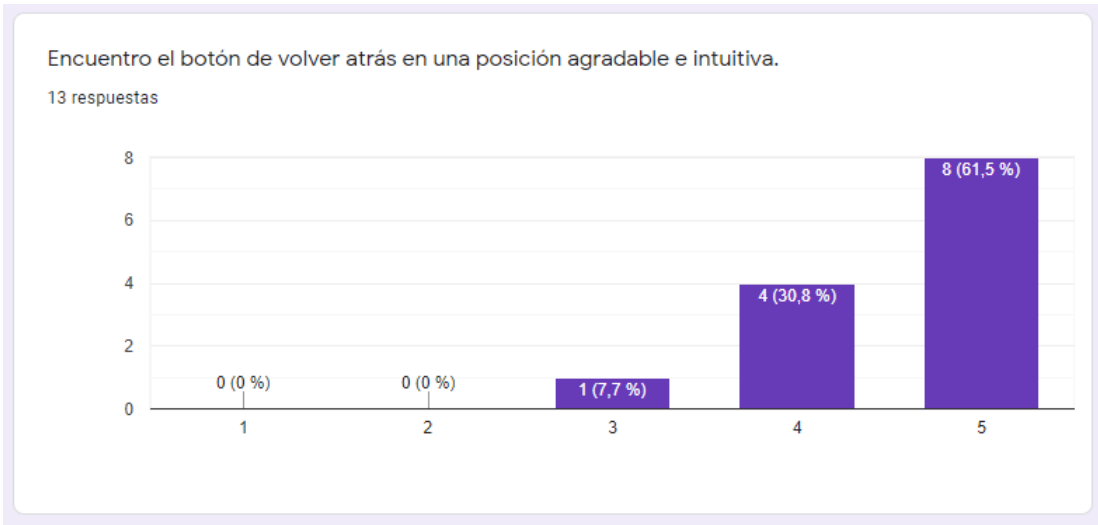


Figura H.24: Resultado de la pregunta 21 del cuestionario de usabilidad

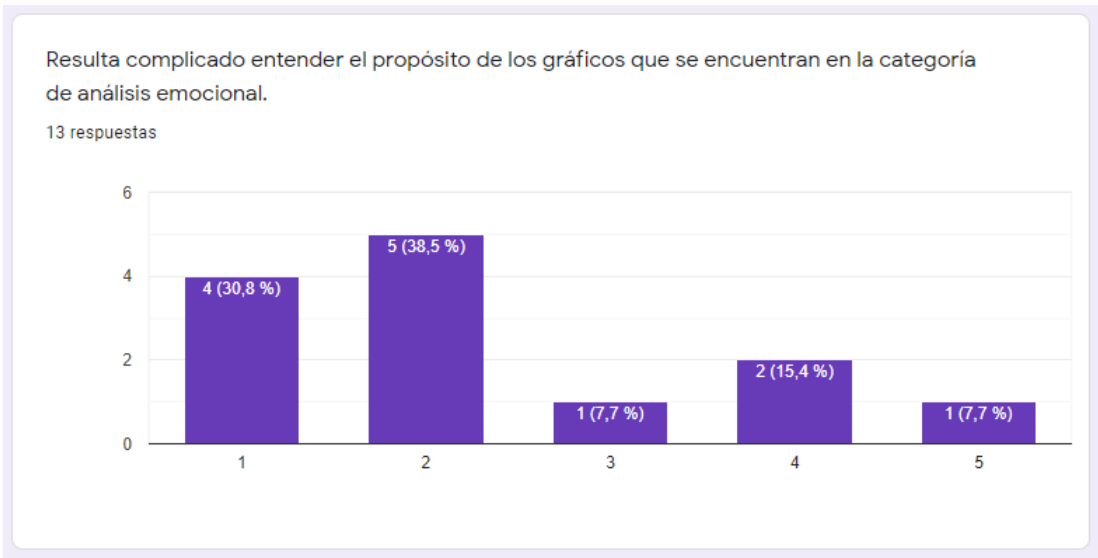


Figura H.25: Resultado de la pregunta 22 del cuestionario de usabilidad

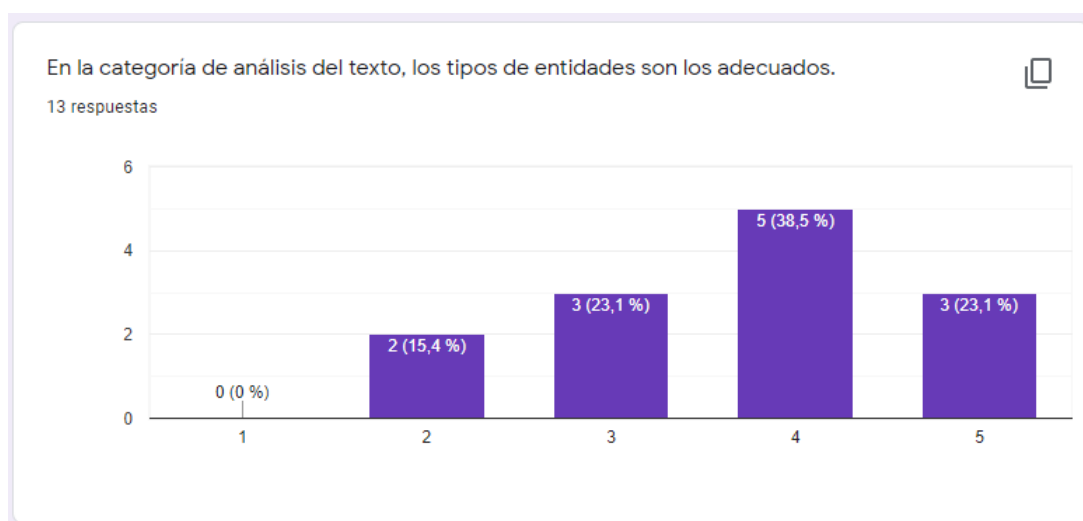


Figura H.26: Resultado de la pregunta 23 del cuestionario de usabilidad

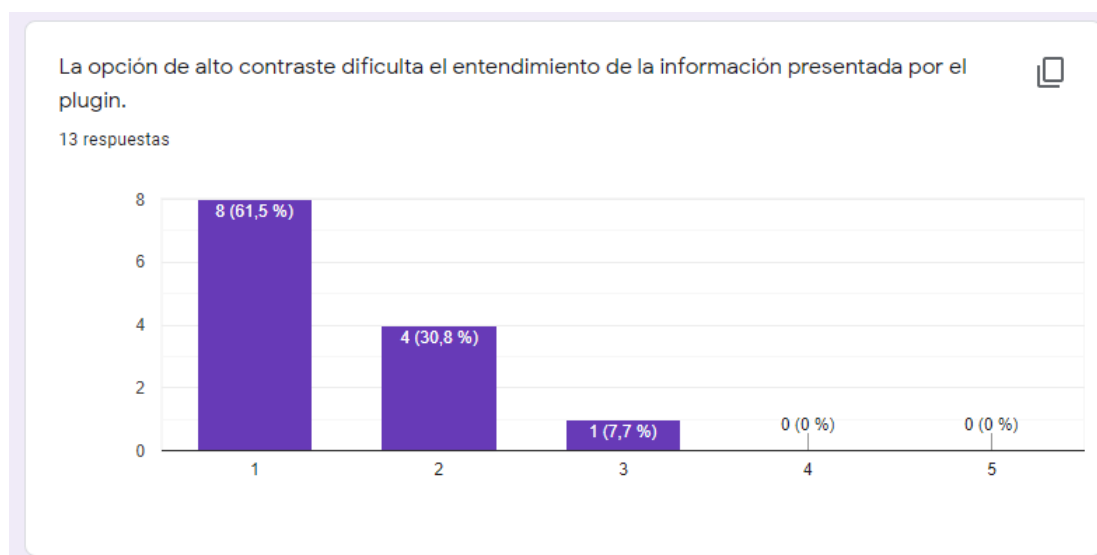


Figura H.27: Resultado de la pregunta 24 del cuestionario de usabilidad

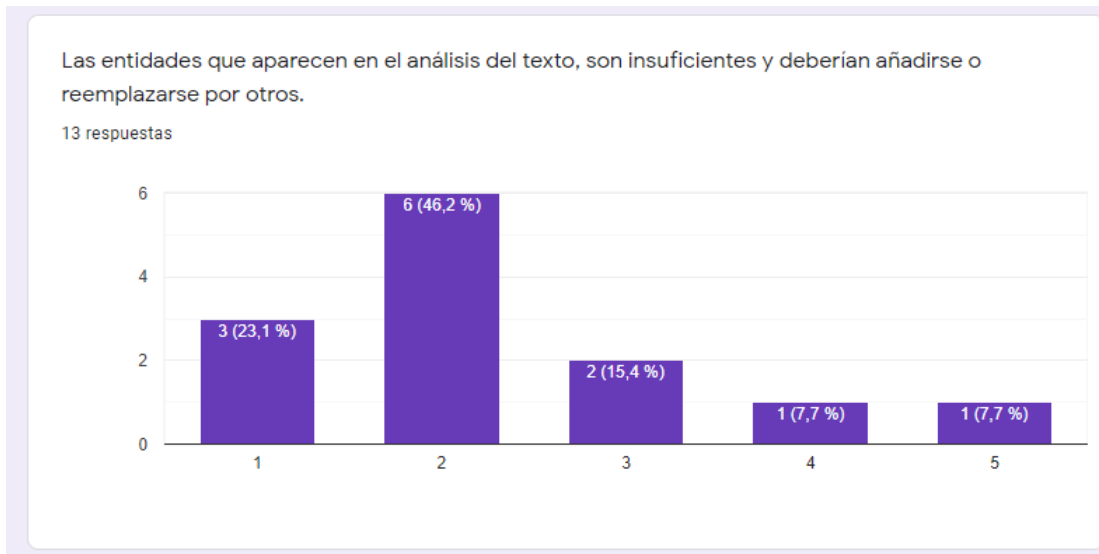


Figura H.28: Resultado de la pregunta 25 del cuestionario de usabilidad

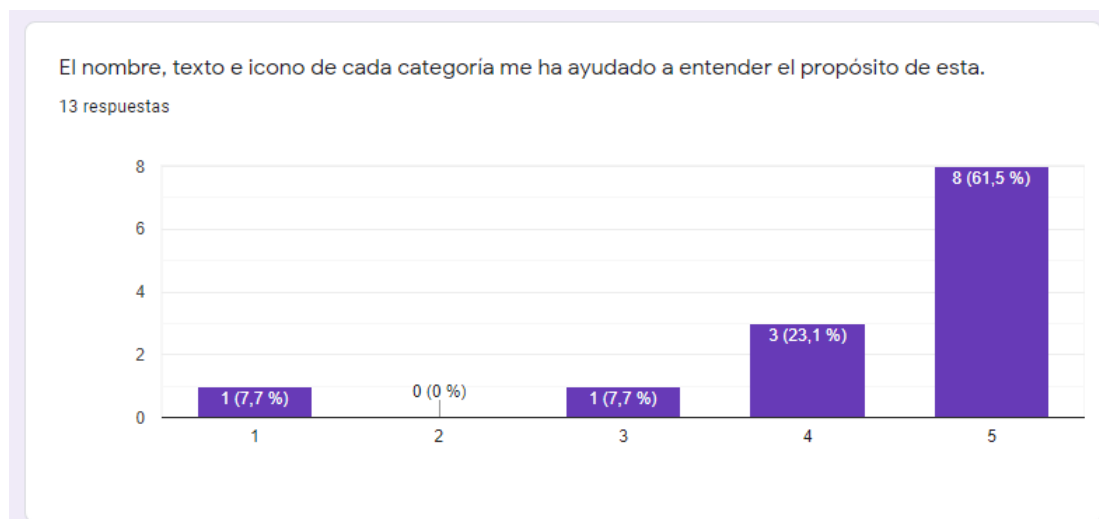


Figura H.29: Resultado de la pregunta 26 del cuestionario de usabilidad

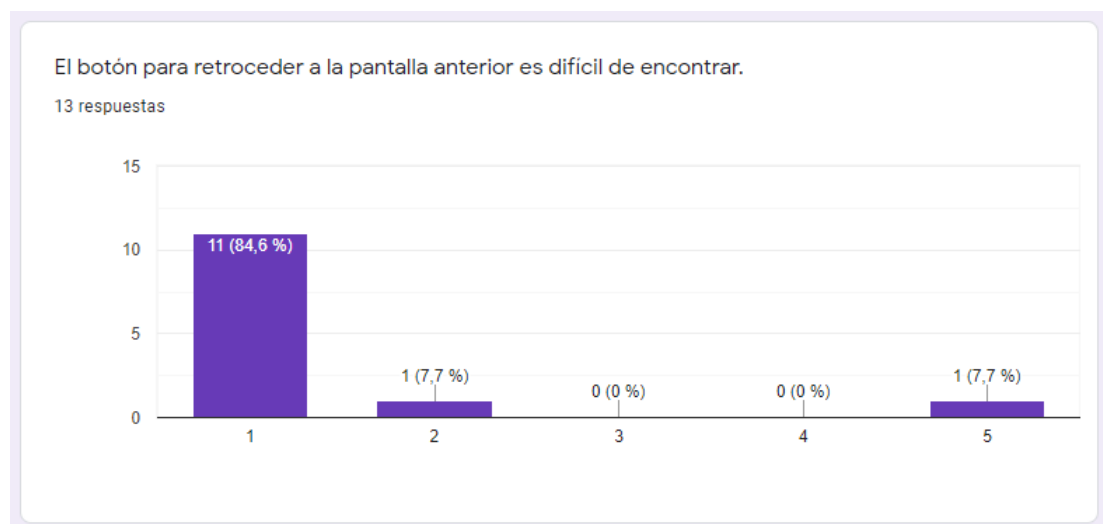


Figura H.30: Resultado de la pregunta 27 del cuestionario de usabilidad

